Uncorrelated Encounter Model of the National Airspace System Version 2.0

A.J. Weinert
E.P. Harkleroad
J.D. Griffith
M.W. Edwards
M.J. Kochenderfer

19 August 2013

Lincoln Laboratory

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LEXINGTON, MASSACHUSETTS

The PEO AV-UAS-USAIC public affairs office has reviewed this report, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

Gary Tutungian
Administrative Contracting Officer
Enterprise Acquisition Division

Massachusetts Institute of Technology
Lincoln Laboratory

Uncorrelated Encounter Model of the National Airspace System Version 2.0

A.J. Weinert
E.P. Harkleroad
J.D. Griffith
M.W. Edwards
M.J. Kochenderfer
Group 42

Project Report ATC-404

19 August 2013

Lexington   Massachusetts

This page intentionally left blank.

ABSTRACT

Airspace encounter models, which provide realistic close encounter situations representative of the airspace, are a critical component in the safety assessment of sense and avoid (SAA) systems. Of particular relevance to Unmanned Aircraft Systems (UAS) is the potential for encountering general aviation aircraft that are flying under Visual Flight Rules (VFR) and which may not be in contact with air traffic control. In response to the need to develop a model of these types of encounters, Lincoln Laboratory undertook an extensive radar data collection and modeling effort involving over 200 radars across the United States. This report describes the structure, content, and usage of that encounter model. The model is based on the use of Bayesian networks to represent relationships between dynamic variables and to construct random aircraft trajectories that are statistically similar to those observed in the radar data. The result is a framework from which representative intruder trajectories can be generated and used in fast-time Monte Carlo simulations to provide accurate estimates of collision risk.

The model described in this report covers only uncorrelated encounters. An encounter with an intruder that does not have a transponder, or between two aircraft using a Mode A beacon code of 1200 (VFR), is uncorrelated in the sense that it is unlikely that there would be prior intervention by air traffic control. The uncorrelated model described in this report is based on data from transponder-equipped aircraft using a 1200 (VFR) Mode A code observed by radars nationwide. As determined from a previous comparison against primary-only tracks, in addition to representing cooperative VFR-on-VFR encounters, this model is representative of encounters between a cooperative aircraft and conventional noncooperative aircraft similar to those using a 1200 transponder code.

The uncorrelated model described in this report is one of several developed by Lincoln Laboratory; the other Lincoln Laboratory encounter models are described in separate reports. For example, a second uncorrelated model was developed for unconventional aircraft that have different flight characteristics than 1200-code aircraft. A correlated encounter model was developed to represent situations in which it is likely that there would be air traffic control intervention prior to a close encounter; the correlated model applies to intruders using a discrete, non-1200 code. Also, a model covering international airspace was developed.

Separate electronic files are available from Lincoln Laboratory that contain the statistical data required to generate and validate uncorrelated encounter trajectories. Details on how to interpret the data file and an example of how to randomly construct trajectories are provided in Appendix A and Appendix B, respectively. A Matlab software package is also available to generate random encounter trajectories based on the data tables.

This report includes updates to the scope and data processing methods used for the version 1.0 uncorrelated model described in [1]. The updates are motivated by increasing interest in SAA applications related to self-separation in addition to collision avoidance. Self-separation encounter models are likely to have stricter requirements, including the need to simulate longer encounters and greater look-ahead time. This report presents a method of defining the model validity length (MVL) requirement for an uncorrelated self-separation encounter model; the MVL is the estimated time horizon over which simulated tracks can be accurately and realistically propagated. This report also covers data processing steps which increase the MVL to approximately 300 seconds. These additional processing steps are designed to remove anomalies both present in the historical data and introduced during processing steps. The new steps discard small portions of the historical radar data and remove additional types of outliers.

Another enhancement to the model is an additional discrete variable specifying the geographic location, which allows a single model to provide specialized coverage of different geographic regions. These regions include the contiguous United States, Alaska, islands, and offshore areas.

TABLE OF CONTENTS

This page intentionally left blank.

LIST OF ILLUSTRATIONS

LIST OF TABLES

1. INTRODUCTION

A key challenge to integrating unmanned aircraft into the National Airspace System (NAS) is development of their ability to sense and avoid local air traffic. Improved aircraft avoidance systems could provide an additional layer of protection that maintains or enhances the current exceptional level of aviation safety and could enable freer access of Unmanned Aircraft Systems (UAS) to the NAS. However, due to their safety-critical nature, rigorous assessment is required before sufficient confidence can exist to certify avoidance systems for operational use. Evaluations typically include flight tests, operational impact studies, and simulation of millions of traffic encounters that explore the avoidance system's performance. Key to these simulations are encounter models that describe the statistical makeup of the encounters in a way that represents what actually occurs in the airspace. Each encounter generated from the model specifies the initial positions and orientations of the aircraft involved and the nominal dynamic maneuvers that would occur without intervention by an avoidance system—in other words, a geometric, dynamic situation which an avoidance system is expected to resolve safely. Identical encounter situations are typically simulated with and without an avoidance system to quantify relative system benefits. Knowledge of the rates at which encounter situations occur in the NAS is used to estimate the rate of near mid-air collisions per flight hour and thus arrive at an overall risk metric.

In these models, encounter situations are abstracted in the sense that there is no consideration of an explicit location or local airspace structure (e.g., airways, metering fixes, approach paths). Rather, the encounters represent what is statistically expected to occur over the lifetime of a given system. If desired, a particular geographic region, altitude layer or airspace class can be specified so that simulated aircraft behavior is representative of that particular situation. Additionally, the flight path of one aircraft can be constrained to focus, for instance, on a particular departure profile or flight condition.

The model described in this report covers approximately 60 to 300 seconds (1 to 5 minutes) before closest point of approach and so is not appropriate for large-scale air traffic impact studies—for example, examination of sector loading or conflict rates. The focus here includes two types of short-term resolution situations:

1. loss of separation has already occurred between two aircraft and collision avoidance becomes paramount and

2. a potential conflict requires a self-separation maneuver, but no loss of separation has occurred and a collision is not imminent.

The model described in this report extends the 1-minute time horizon of the version 1.0 model described in [1] to support modeling of self-separation applications in addition to collision avoidance applications.

One system that has been rigorously tested using encounter models is the Traffic alert and Collision Avoidance System (TCAS). As part of the TCAS certification process in the 1980s and 1990s, several organizations tested the system across millions of simulated close encounters and

evaluated the risk of a near mid-air collision (NMAC, defined as separation less than 500 ft horizontally and 100 ft vertically) [2–5]. This analysis ultimately led to the certification and U.S. mandate for TCAS equipage on larger transport aircraft. More recently, Eurocontrol and the International Civil Aviation Organization (ICAO) performed similar sets of simulation studies for European and worldwide TCAS mandates [6,7]. Note that TCAS is certified specifically for collision avoidance— not for self-separation.

The design of an avoidance system represents a careful balance to enhance safety while ensuring a low rate of unnecessary maneuvers. This balance is strongly affected by the types of encounter situations to which the system is exposed. It is therefore important that simulated encounters are representative of those that occur in the airspace. Hence, tremendous effort has been made by various institutions since the early 1980s to develop encounter models based on radar data [2,4,8–11]. The primary contribution of this report and [1] is an approach to encounter modeling that is based on a Bayesian statistical framework and which uses recent radar data collected across the United States. The advantage of applying a Bayesian framework is that it allows optimal leverage of available radar data to produce a model that is representative of actual operations and can simulate encounters not explicitly present in the radar data. The result is a framework from which representative intruder trajectories can be generated and used in fast-time Monte Carlo simulations to provide accurate estimates of collision risk.

A byproduct of the encounter modeling effort was the development of aircraft track and traffic density databases of the National Airspace System. Example plots of traffic density data are provided in [1].

1.1 ENCOUNTER TYPES

The encounters covered by this model involve aircraft in the final few minutes before a collision. It is assumed that prior safety layers (e.g., airspace structure, Air Traffic Control (ATC) advisories or vectors) have failed to maintain standard separation distances between aircraft. The model is therefore useful in describing the types of situations that need to be addressed by an avoidance system, but will not address other separation aspects such as ATC communications or coordination.

Because they are by far the most likely to occur, only pairwise (two-aircraft) encounters are explicitly discussed in this model. If required, the probability of multiple aircraft encounters can be determined from the traffic density database. Random multi-aircraft encounters can be constructed by combining three or more trajectories generated from the model presented here.

There are two fundamental types of encounters: correlated and uncorrelated. In the first, both aircraft involved are cooperative (i.e., have a transponder) and at least one is in contact with air traffic control. It is then likely that at least one aircraft will receive some notification about the traffic conflict and begin to take action before an avoidance system becomes involved. This type of encounter is called "correlated" because the trajectories of each aircraft may involve maneuvers that are correlated to some degree due to this prior intervention. The second type of encounter is called "uncorrelated" and involves at least one noncooperative aircraft (i.e., not using a transponder) or two aircraft flying under Visual Flight Rules (VFR) without flight following (i.e.,

using a transponder Mode A code of 1200). In these encounters, it is unlikely that air traffic control would become involved prior to the close encounter; rather the two aircraft must rely solely on visual acquisition (or some other avoidance system) at close range to remain separated. Such encounters tend to be uncorrelated since there is no coordinated intervention prior to the close encounter: the assumption is that the two aircraft blunder into close proximity. A complete evaluation of unmanned systems will require analysis using both correlated and uncorrelated models.

This report focuses on uncorrelated encounter modeling; it describes a model based on beacon reports from aircraft squawking 1200, not radar returns from noncooperative traffic. Radar surveillance of noncooperative targets is complicated due to clutter and missed detections, making identification of real tracks difficult. The lack of a transponder means that only position information, and no identity code or altitude, is available[1]. Hence it is difficult to infer vertical rates, an important component of the encounter model.

Beacon-equipped aircraft can transmit either a discrete Mode A code or the non-discrete code of 1200. Aircraft using code 1200 tend to be general aviation flying VFR. They generally fly at low altitudes and make significantly more maneuvers than transport aircraft, both horizontally and vertically. Thus to a large degree they resemble noncooperative aircraft. Due to the poor quality of noncooperative data, 1200-code tracks serve as surrogates for primary-only tracks in the construction of this model; true noncooperative tracks are not used in this model.

Previous work shows that 1200-code tracks are a proper surrogate for certain classes of noncooperative traffic in the NAS [1], but they are not suitable for all categories of noncooperative targets. For example, most balloons, ultralights, and gliders do not fly like transponder-equipped aircraft squawking 1200 and hence a different data source is required for these aircraft [12]. Additionally, where data are not available, a model can be manually constructed based on knowledge of typical flight trajectories and performance limits.

1.2 MODEL SELECTION

UASs are envisioned to operate in a variety of environments. As an example, the Triton UAS will operate over the contiguous United States, in offshore environments, and in oceanic environments [13]. To address this variety of situations, Lincoln Laboratory has developed a variety of encounter models to evaluate SAA systems. There are four types of encounter models:

- **Uncorrelated Encounter Model of the National Airspace System (U):** An uncorrelated encounter model is used to evaluate the performance of SAA systems when at least one aircraft is noncooperative or neither aircraft is in contact with air traffic control (ATC) [1]. This model was developed from 1200-code aircraft observed in the NAS and now includes the offshore environment out to the limits of radar coverage [14].

- **Correlated Encounter Model of the National Airspace System (C):** A correlated encounter model is used to evaluate the performance of SAA systems when both aircraft are

---

[1]Some military radars have height-finding capability, although the accuracy of the altitude generated is significantly inferior to the transponder Mode C altitude.

3

cooperative and at least one aircraft is receiving ATC services [15]. This model was developed from observed encounters in the NAS and also includes the offshore environment.

- **Encounter Models for Unconventional Aircraft (X):** This encounter model is used to evaluate the performance of a SAA system when encountering unconventional aircraft, defined as aircraft unlikely to carry a transponder [12]. Examples of unconventional aircraft include balloons, blimps, gliders, and skydivers. This model was developed from GPS-recorded tracks that are posted online.

- **Due Regard Encounter Model (D):** This encounter model is used to evaluate SAA systems for UAS flying due regard in oceanic airspace. This model is primarily built from self-reported positions of aircraft flying in oceanic airspace.

Each of these encounter models includes variables that account for variations in encounters with respect to different types of airspace. For example, one of the variables in the uncorrelated encounter model is Airspace Class, which includes B, C, D, and O (Other). Table 1 indicates the appropriate model to use based on the study being performed[2]. For the offshore environment, the correlated and uncorrelated encounter models encompass encounters more than 1 NM beyond the shore and the due regard model begins at 12 NM, where due regard flight is permitted[3]. The oceanic environment includes international airspace beyond radar coverage. Note that no existing model covers encounters between two IFR aircraft in oceanic airspace. The reason for this is that one cannot observe encounters of sufficient fidelity in the available data feeds. Similarly, there is no model that covers encounters with visual flight rules (VFR) or noncooperative aircraft in oceanic airspace due to a lack of surveillance data. If models of these types of encounters are required, they should be built using the best available assumptions about aircraft behavior and should leverage data from similar encounter models as is necessary. For example, one could use offshore models or a subset of the encounters covered by contiguous United States (CONUS) models.

Only the uncorrelated conventional model (U) is discussed in this report; the other models are described in other reports.

1.3   RADAR DATA

The radar data used to build the model comes from the 84th Radar Evaluation Squadron (RADES) at Hill AFB, Utah. RADES receives radar data from FAA and Department of Defense sites throughout the United States. They maintain continuous real-time feeds from a network of radars, including long-range ARSR-4 radars around the perimeter of the United States and short-range ASR-8, ASR-9, and ASR-11 radars in the interior. Radar ranges vary from 60 to 250 NM. In addition to CONUS coverage, the latest encounter model now includes radar coverage of Alaska, the Aleutian Islands and the surrounding waters; Hawaii; and the Lucayan Archipelago and Greater Antillles

---

[2]Note that a model does not exist for combinations left blank.

[3]Note that one cannot build a *correlated* encounter model from radar data for due regard flight in the offshore environment because one does not observe a sufficient number of encounters between instrument flight rules (IFR) and non-IFR traffic beyond 12 NM from the shore.

TABLE 1

Encounter model categories.

Aircraft of Interest		Intruder Aircraft			
Location	Flight Rule	IFR	VFR	Noncooperative Conventional	Noncooperative Unconventional
CONUS	IFR	C	C	U	X
	VFR	C	U	U	X
Offshore	IFR	C	C	U	X
	VFR	C	U	U	X
	Due Regard	D	U	U	X
Oceanic	IFR				
	VFR				
	Due Regard	D			

in the Caribbean. Figure 1 shows the approximate coverage by the 229 radars that were used to construct the model.

To provide the necessary information to model aircraft trajectories, four weeks of 1200-code aircraft reports were collected from 229 radars covering the time periods 3–9 March 2010, 3–9 June 2010, 3–9 September 2010, and 3–9 December 2010; the sample includes approximately 2.3 million aircraft tracks and over 295,000 flight hours after all processing steps. As discussed in [1], two or more weeks of data provide sufficient statistical power and generality to enable a valid and representative model of 1200-code flight characteristics. In addition to this focused four-week data set, aircraft tracks from larger sets of radar data are archived and available for traffic density studies as needed.

Note that there are a number of advantages to the RADES data feed compared to the Enhanced Traffic Management System (ETMS) data often used in airspace analyses. ETMS data include only cooperative aircraft on filed IFR flight plans and provide updates once per minute showing aircraft position after processing by air traffic control automation. In contrast, the RADES data feed streams continuously and directly from the radar, including primary-only radar returns as well as cooperative transponder returns (whether on a flight plan or not), providing track updates every 5 or 12 s without being affected by automation systems. These properties ensure that the filters and trackers have the best raw data with which to begin processing.
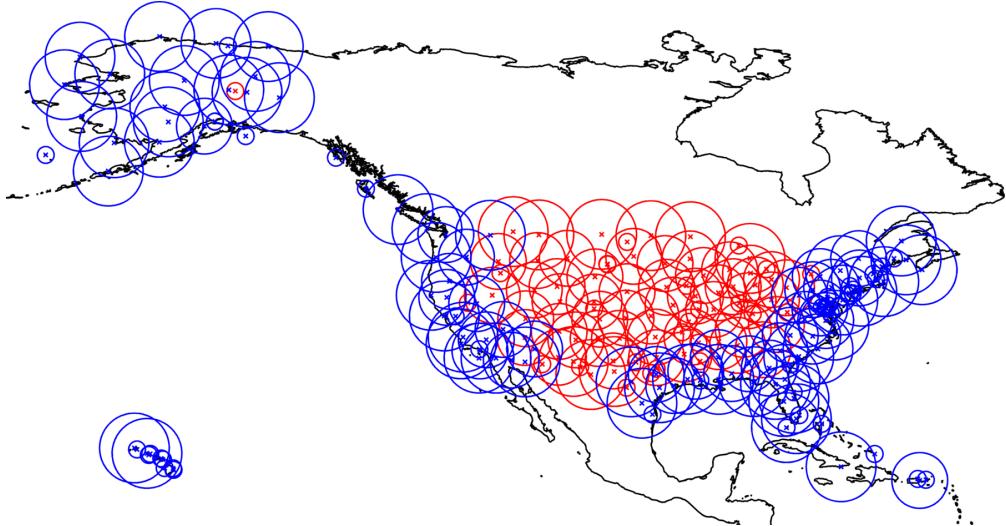
Figure 1. RADES radar coverage map. Radars in red only provide data over CONUS.

1.4 PROCESS OVERVIEW

Figure 2 diagrams the steps involved in processing radar data to build the encounter model and generate encounters for simulation. The high-level approach is to model nominal aircraft trajectories using Markov models represented by Bayesian networks (Section 2). The first step in constructing a Markov model involves extracting features, such as turn rate and vertical rate, from the radar data. From these features, sufficient statistics[4] are collected to describe the distribution over maneuvers and other properties of trajectories (Section 3). Bayesian model selection methods are used to search for the network structure that best represents the observed data (Section 4). The best network structure is then selected and the associated sufficient statistics are obtained to generate new trajectories that are representative of those observed by radar (Section 5). Finally, the trajectories are used in a dynamic simulation to evaluate encounters between aircraft with or without an avoidance system (Section 6). Section 7 discusses using the encounter model for large-scale safety analysis and collision risk estimation.

A series of Appendices is also included to provide additional detail and background data.

A digital representation of the sufficient statistics and model structure described in this report are available on request from MIT Lincoln Laboratory. An example of using the data in that file to construct a random trajectory is provided in Appendix B. Additionally, a Matlab software package is available to generate random trajectories using the data tables.

This report uses standard aviation units listed in Table 2.

---

[4]The conditional distributions of feature counts are called *sufficient statistics* because they provide a summary of the data that is sufficient to compute the posterior distribution from the prior. For an introduction to Bayesian statistics, see [16].
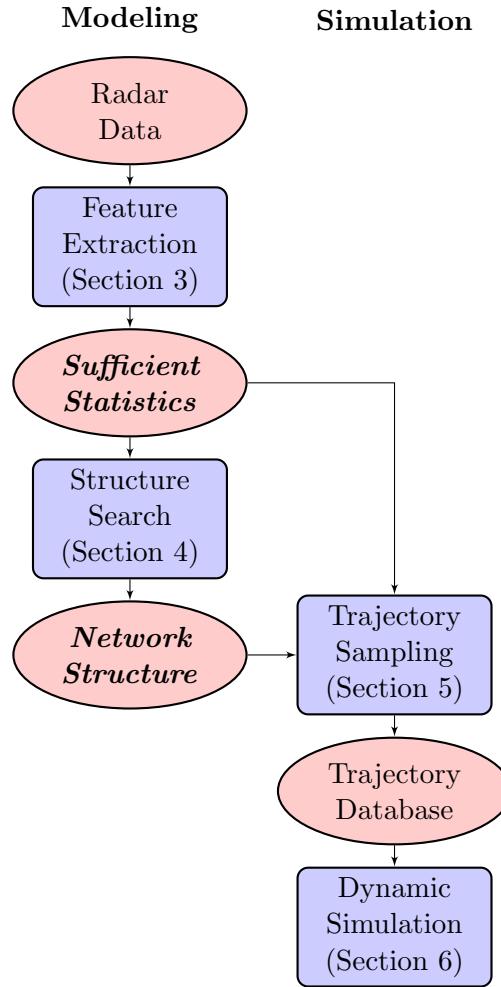
Figure 2. Modeling and simulation process overview. The sufficient statistics and network structure (emphasized) are the main elements provided as part of this work.

TABLE 2

Encounter model units.

Quantity	Units
altitude	ft
position coordinates	NM
time	s
vertical rate	ft/min
turn rate	deg/s
airspeed (true)	kt
airspeed acceleration	kt/s

This page intentionally left blank.

2. MODEL

The model represents nominal flight—flight without avoidance maneuvering—using a Markov process and a dynamic Bayesian network data structure. A Markov process is a stochastic process in which the probability distribution over future states is conditionally independent of past states given the present state. In other words, only the present state is needed to predict the next state.

The states in the model specify how an aircraft's position, altitude, and airspeed change over time. In particular, each state specifies a vertical rate $\dot{h}$, turn rate $\dot{\psi}$, and airspeed acceleration $\dot{v}$. Given an initial airspeed v, position (x, y, h), heading ψ, vertical rate $\dot{h}$, geographic location G, altitude layer L, and airspace class A, one can determine from the model how the aircraft trajectories evolve over the course of the encounter.

One way to represent a Markov model is with an exhaustive state-transition matrix that specifies the probability of transitioning between all pairs of states. Unfortunately, the number of independent parameters—independent probabilities—required to define the matrix grows super-exponentially with the number of variables defining the model. The more independent parameters in the model, the more data needed to properly estimate their values. However, dynamic Bayesian networks can leverage conditional independence between some variables to greatly reduce the number of parameters. The structure of the dynamic Bayesian network can be learned by maximizing the posterior probability of the network structure given the data.

Appendix F provides the necessary background on Bayesian networks. The remainder of this section defines an encounter in further detail, introduces the variables used to describe an encounter, and presents the modeling approach.

2.1 MODEL VARIABLES

There are seven variables in the uncorrelated encounter model:

- **Geographic location** G**:** Aircraft behavior may vary across different geographic regions due to varying weather patterns, fleet mix, navigation equipage, regulations, mission types, and other factors. Including a variable denoting the geographic region allows the model to be customized to the different environments in which aircraft operate; it may take one of four values:

  1. Contiguous United States (CONUS), Alaska, Canada, and Mexico,
  2. Islands,
  3. CONUS Offshore, and
  4. Islands Offshore.

  This variable is an addition to the model which was not included in the version 1.0 model [1].

- **Airspace class** A**:** The airspace class variable accounts for the variation in aircraft behavior across different airspace classes. The allowed classes include B, C, and D as defined by the

Federal Aviation Administration (FAA). The fourth value O represents "other" airspace, including airspace Classes E and G. Some airspace covered by the model is not regulated by the FAA. There should be no VFR aircraft in Class A airspace due to its requirement of Instrument Flight Rules.

- **Altitude layer** L**:** Airspace is divided into four altitude layers in a process similar to prior encounter models developed by Eurocontrol. The first layer spans 500 to 1200 ft Above Ground Level (AGL) to capture aircraft in the traffic pattern or performing low-level maneuvers. The second layer spans a transition zone from 1200 to 3000 ft AGL—the cruise altitude where the hemispheric rule begins. The third layer spans 3000 ft AGL to 5000 ft AGL and covers a mix of low-altitude en route and maneuvering aircraft. The fourth layer spans 5000 ft AGL to 18,000 ft and should cover most en route VFR traffic.

- **Airspeed** v**:** Airspeed is permitted to change every second during flight; this variable represents true airspeed. When estimating airspeed from the radar data, the wind speed is ignored such that the airspeed is assumed equal to the ground speed.

- **Acceleration** $\dot{v}$**:** Acceleration is permitted to change every second, providing higher-fidelity motion than prior models.

- **Turn rate** $\dot{\psi}$**:** Turn rate is permitted to change every second. Prior European and ICAO models allowed only a single turn during an encounter.

- **Vertical rate** $\dot{h}$**:** Vertical rate is permitted to change every second. All prior cooperative models allowed only a single vertical acceleration period during an encounter.

Section 3.7 provides more details on each variable.

Because many of the variables are closely related due to physical constraints and flight characteristics (e.g., turn rate and vertical rate) it is important to properly represent correlations in the model. Independently sampling from distributions for turn rate and vertical rate would miss these important relationships. The remainder of this section explains how to model joint probability distributions over these variables to ensure proper consideration of correlations. To generate an encounter, first randomly sample from the joint distribution over the encounter variables to define the encounter geometry and initial conditions. Second, use a Markov model to determine how the dynamic variables—such as turn rate, vertical rate, and airspeed acceleration—evolve during the course of the encounter. There are two corresponding separate probability distributions in the model: an initial distribution to set up an encounter situation and a transition distribution to describe how the dynamic variables specifying the trajectories of the aircraft evolve over time.

2.2 INITIAL DISTRIBUTION

The aircraft encounter model represents the distribution over the initial values of $\dot{h}$, $\dot{\psi}$, $\dot{v}$, v, L, G, and A as well as the time-varying history of $\dot{h}$, $\dot{\psi}$, and $\dot{v}$ during the course of an encounter. Probabilistic relationships between these variables are represented using a Bayesian network. An example

of such a relationship is the one between turn rate and vertical rate. Without properly capturing this dependency and other important dependencies, unrealistic situations may be generated—for example, involving aircraft with simultaneously high climb rates and high turn rates. Initial position and altitude is determined through a separate process described in Section 6.

A Bayesian network (e.g., Figure 3a) includes a series of nodes represented by rectangles and directed arcs or edges represented by arrows. Each node corresponds to a particular variable that may take on one of several discrete values with associated probabilities. Certain variables, such as airspace class or altitude layer, are naturally quantized into a few discrete values such as B, C, D, and O for airspace class. Continuous variables, such as vertical rate or turn rate, are described by a series of discrete bins from which a continuous value is later selected. Within each node, then, is a description of the possible values a variable can take and the probability that each value will occur. The directed arcs describe how the probabilities of one variable depend on other variables. Arrows leading into a particular node denote which parent nodes must first be evaluated in order to select the value of the given node. For example, referring to Figure 3a, the probabilities for node L depend on the values of node G and node A; the probabilities for node v depend on the values of nodes G, A and L. In the latter case, for instance, there would be a conditional probability table describing the probability of each possible value of v jointly conditioned on the values of G, A and L: $P(v \mid G, A, L)$.

Because there are many possible ways to connect variables in the model, it is necessary to use a quantifiable scoring process to evaluate the quality of each candidate network. The design of this model uses a Bayesian scoring method (Appendix F) to evaluate different Bayesian network structures and choose a structure that optimizes the representation of the observed trajectories; Section 4 describes this process. Figure 3a shows the optimized structure for the initial distribution.

Given this optimized structure, sufficient statistics extracted from data, and a Bayesian prior, the Bayesian network is sampled to construct representative combinations of initial conditions: geographic location, airspace class, altitude layer, vertical rate, turn rate, airspeed, and acceleration; Section 5 describes the sampling process. The nodes and directed arcs in the structural diagram show the order in which this sampling occurs. For example, based on the structure in Figure 3a, to determine the initial state of the aircraft, first randomly determine a geographic location G. Once the geographic location has been determined, an airspace class A is selected. The probabilities associated with each airspace class depend on which geographic location was chosen earlier. Once G and A have been selected, the next step is to randomly select altitude layer L, and so on. Appendix B provides an example of working through the process. An alternative is to fix a geographic location and airspace class for a particular study and then randomly select values for the remaining variables.

2.3 TRANSITION DISTRIBUTION

A separate Bayesian network models how the variables $\dot{h}$, $\dot{\psi}$, and $\dot{v}$ evolve over time. In this network, the first layer represents the state of the system at the present time step and a second layer represents its state at the next time step. There are dependencies between layers. Such a

two-layer temporal Bayesian network is known as a dynamic Bayesian network [17, 18]. Parameter and structure learning in dynamic Bayesian networks is similar to that for other Bayesian networks (Appendix F).

Figure 3b shows the structure used for the transition network. As with the initial distribution, it is the highest-scoring structure among a set of candidate network structures (see Section 4). Given a structure, sufficient statistics extracted from data, and a prior distribution, sampling from the Bayesian network determines the next vertical rate, turn rate, and airspeed acceleration command.

In general, time steps in dynamic Bayesian networks may be of any duration, but this model uses steps of 1 second. Shorter time steps allow more frequent variations in airspeed, vertical rate, and turn rate, but they require more computation per unit of simulation time. Time steps of 1 second are appropriate given typical timescales of dynamic maneuvers; this choice provides a reasonable compromise between maneuver complexity and computational cost concerns.

A complete trajectory is constructed by updating the aircraft state in 1-second intervals. Within each interval, the three derivative variables $\dot{h}$, $\dot{\psi}$, and $\dot{v}$ are treated as target values and held constant. A dynamic model is used to compute and update the aircraft state at each time step based on these piecewise-constant target values. The dynamic model is independent of this encounter model and not discussed in this report, but Appendix D describes a process for validating an encounter model against the one used at Lincoln Laboratory.
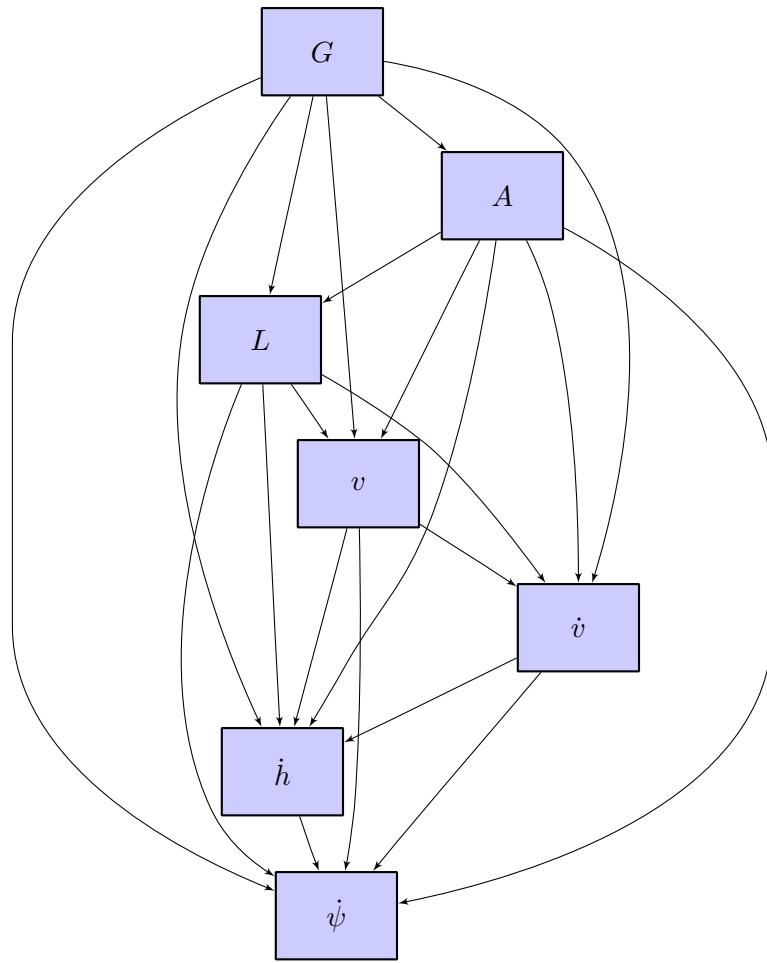
(a) Initial distribution.
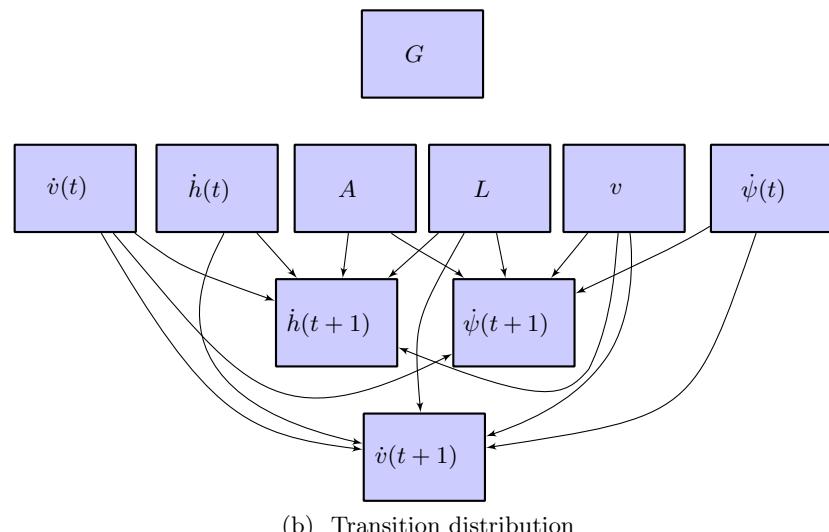
(b) Transition distribution

Figure 3. Bayesian networks representing the variable dependency structure for the initial and transition distributions.

13

This page intentionally left blank.

3. ESTIMATION

At a high level, the modeling process involves taking in a large volume of radar data and carefully filtering and processing that data to extract features of aircraft trajectories. Features include static variables that specify an encounter (such as altitude layer or initial airspeed) and multiple, dynamic variables that describe aircraft motion leading up to and through the closest point of approach (such as turn rate, vertical rate, and airspeed acceleration every second). To aid in data processing, each feature was quantized into several bins and counts were taken of the frequency with which each bin was occupied by radar data. Based on these counts, or sufficient statistics, probability tables were then constructed so that each feature can be randomly generated such that the overall geometries and dynamics are representative of the actual events observed in the radar data.

Accordingly, the inputs to this process are raw radar reports (range, azimuth, altitude, time) and the outputs are conditional probability tables specifying the likelihood that a given feature will take on a value within a bin corresponding to each table cell. This section describes the process which transforms radar tracks into sufficient statistics that may be used to model uncorrelated encounters. Figure 4 outlines the multiple-stage feature extraction process.

The raw radar data are first processed using a tracking algorithm developed at Lincoln Laboratory [19]. A correlation algorithm, also developed at Lincoln Laboratory [20], then correlates tracks from multiple radars to produce a global air picture of all the tracks in the airspace of interest (see Appendix E). Tracks with fewer than ten scans are eliminated because approximately ten scans are required to accurately estimate the various maneuver rates. Tracks are also eliminated if any of their associated reports lie inside Special Use Airspace, whose boundaries are defined in the Digital Aeronautical Flight Information File (DAFIF), 8th Edition, managed by the National Geospatial-Intelligence Agency (NGA).

3.1   MISSING VALUES, AIRSPEED, AND VERTICAL RATE OUTLIER REMOVAL

The first step in processing the raw radar tracks is to detect and remove outliers. This step ensures that the tracks used to build the model are physically realistic.

In the horizontal plane, track segments with ground speeds above 600 kt are removed as follows. Assume, consistent with FAA regulations, that aircraft will not approach or exceed Mach 1 (600 kt) at the modeled altitudes. Estimate the speed at each report by dividing the distance between reports by the time interval between them. Identify reports with speed above the 600 kt threshold and record points adjacent to them in a list of candidates for removal. Remove the candidate that minimizes the sum of speeds above the threshold. Repeat the removal process until there are no longer any segments with speeds above the threshold. Observations of larger speeds may be due to sensor noise, aircraft behavior inconsistent with regulations, or other anomalies; removal of these cases helps to calibrate the model more closely to typical aircraft behavior.

In the vertical plane, altitude information is required, so it is necessary to remove missing Mode C altitude reports from consideration. Next, using the same process used for the horizontal
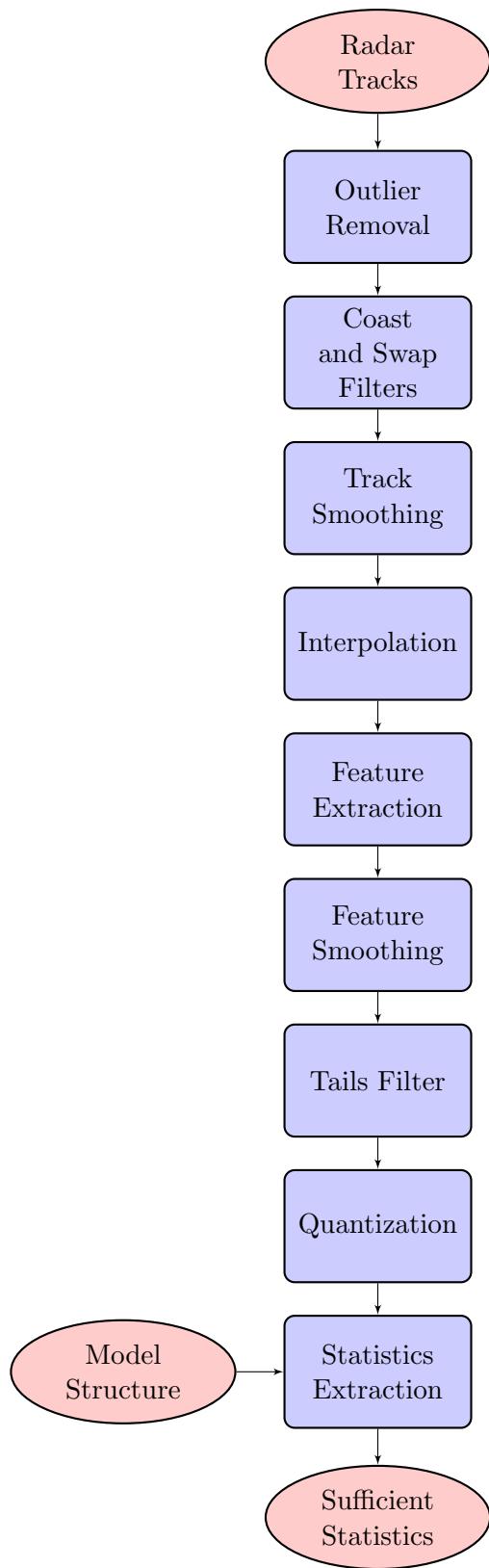
Figure 4. Estimation process flow.

plane, remove outliers with vertical rates greater than $5000\,\text{ft/min}$ or less than $-5000\,\text{ft/min}$. A limit of $\pm 5000\,\text{ft/min}$ is appropriate because it slightly exceeds the reported climb rates achieved by aerobatic race aircraft; observations of larger vertical rates may be due to sensor noise or other anomalies. Next, remove altitude reports that come before the first position report or after the last position report to prevent extrapolation. Because the lowest altitude layer begins at $500\,\text{ft}$, ignore altitude reports below $500\,\text{ft}$. Also, remove position reports before the first altitude report or after the last altitude report—again to prevent extrapolation. As with horizontal outliers, removal of these vertical outliers helps to calibrate the model more closely to typical aircraft behavior.

After all outlier removal steps discussed in this subsection, discard tracks with fewer than ten valid scans—again to ensure there is enough data to accurately estimate the various maneuver rates.

Sections 3.2 and 3.3 discuss additional outlier removal methods which provide higher-confidence results by removing other types of outliers.

3.2   COAST FILTER

Coasting occurs when surveillance of an aircraft is temporarily lost, resulting in long time intervals of missing data relative to radar scan time. In the example in Figure 5, there is a time difference of over $600\,\text{s}$ between two returns resulting in roughly $50\,\text{NM}$ of false interpolated data between the returns. This example illustrates the need for a filter to identify tracks with large gaps and keep only the intervals with contiguous data.
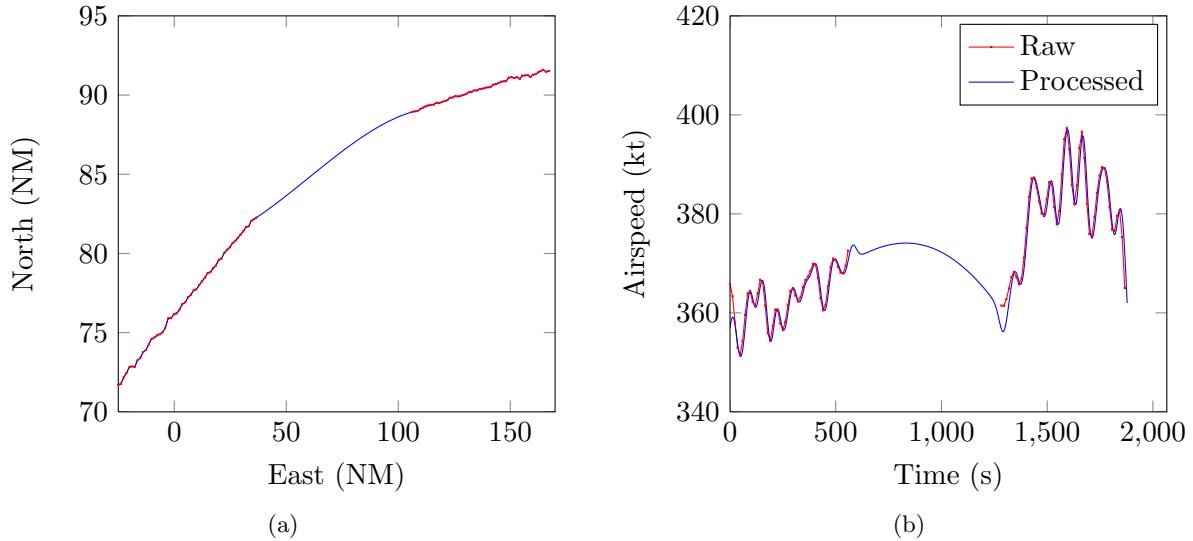
Figure 5. Coasting example.

To accomplish this removal, the coast filter calculates the time between returns and splits the data if the time between returns is greater than three times the median time between returns for the entire track; a track is split at several points if several outliers occur. As an example, for a long range radar such as the ARSR-4, the scan rate is 12 s; if the time between two returns recorded by this type of radar is greater than 36 s, then the track would be split into separate tracks containing only the intervals with contiguous data. This splitting prevents the creation of false interpolated data inside the coasting interval.

3.3 SWAP FILTER

Track swaps occur when a tracker follows one aircraft, then detects another aircraft and continues tracking the second aircraft instead, mistakenly treating them as a single aircraft; they can also be produced by poor position outlier detection. Track swaps are generally identifiable by a distinctive spike of unrealistic velocity as shown in Figure 6, which creates two problems. First, unrealistic and false airspeed data is collected. Second, one track now includes data for two different aircraft which are possibly different aircraft types with different performance characteristics. The risk of this error is greater in terminal airspace where one aircraft may be landing while another is taking off.

(a)

(b)
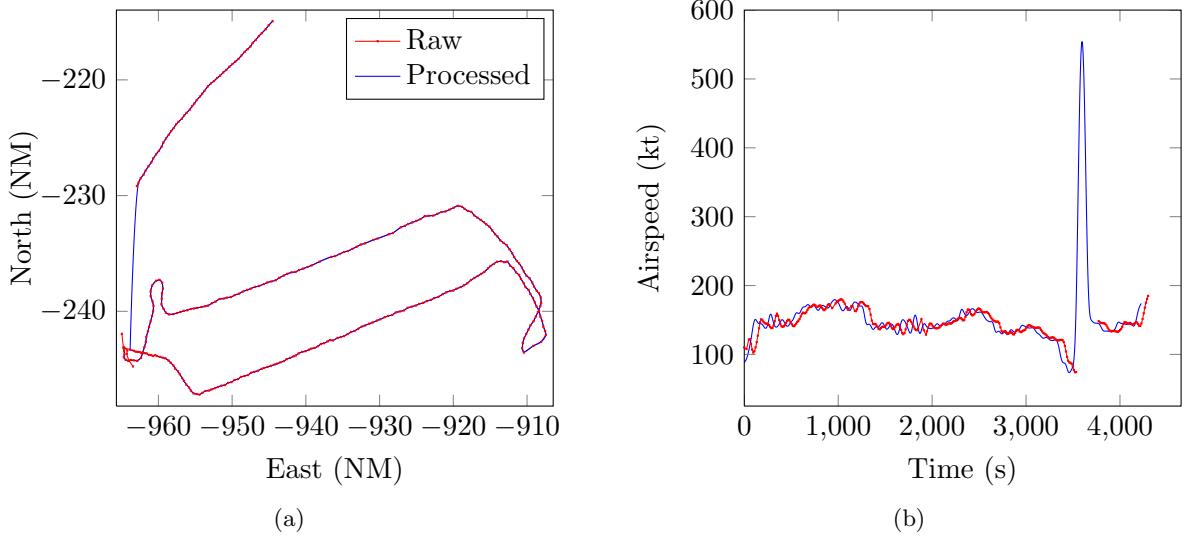
Figure 6. Track swap example.

The swap filter identifies and removes these cases. It begins by calculating the horizontal distance traveled per unit time between radar returns, which gives the instantaneous speed at each point along the track. If any individual speed is greater than 2.1 times the median speed for the entire track, the filter removes this data point and splits the track, keeping only intervals of contiguous data before and after it. As with the coast filter, a track is split at several points

if several outliers occur; this procedure prevents false interpolated data from distorting encounter model statistics.

3.4 TRACK SMOOTHING

After removing any outliers from a track, the remaining data points are smoothed first horizontally, then vertically. The same smoothing method is used in both cases. The following general formula transforms a raw trajectory $(t_1, \boldsymbol{x}_1), \ldots, (t_n, \boldsymbol{x}_n)$ into a smoothed trajectory $(t_1, \boldsymbol{y}_1), \ldots, (t_n, \boldsymbol{y}_n)$:

$$\boldsymbol{y}_i = \frac{\sum_j w(t_i, t_j) \boldsymbol{x}_i}{\sum_j w(t_i, t_j)}, \tag{1}$$

where $w(t_i, t_j)$ is a weighting function that monotonically decreases as the difference between t_i and t_j increases. The weighting function uses the following definition based on a Gaussian kernel with standard deviation σ:

$$w(t_i, t_j) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(t_i - t_j)^2}{2\sigma^2}\right). \tag{2}$$

When smoothing horizontally, use $\sigma = 5\,\mathrm{s}$ and while smoothing vertically, use $\sigma = 15\,\mathrm{s}$. A larger σ is required for vertical smoothing because of 100 ft Mode C quantization. These values for σ were selected after testing different standard deviations on a sampling of horizontal and vertical profiles in the data; the chosen values preserve the underlying tracks while removing noise.

3.5 INTERPOLATION

The time interval between radar scans in the data is much longer than the 1 s time step of the dynamic Bayesian network. Terminal (short range) radars scan aircraft approximately every 5 s, and en route (long range) radars scan aircraft every 10 to 12 s. Additionally, it is common for radars to skip one or more consecutive scans of a target and some scans produce outliers that need to be removed (Section 3.1). Hence, interpolation is required to estimate the parameters in the dynamic Bayesian network. A piecewise-cubic Hermite interpolation scheme is selected because it preserves monotonicity and shape [21].

Figure 7 shows the result of outlier detection, smoothing, and interpolation on an example track from the radar data set. Figure 8 shows the result of piecewise-cubic Hermite interpolation on a different example smoothed track.

3.6 TAILS FILTER

Filtering out and removing the tails of source tracks improves data quality. Specifically, after preprocessing of radar data and feature extraction, this step removes an interval of time at the beginning and end of each track. Removing 60 seconds gives the best result. Though ignoring these
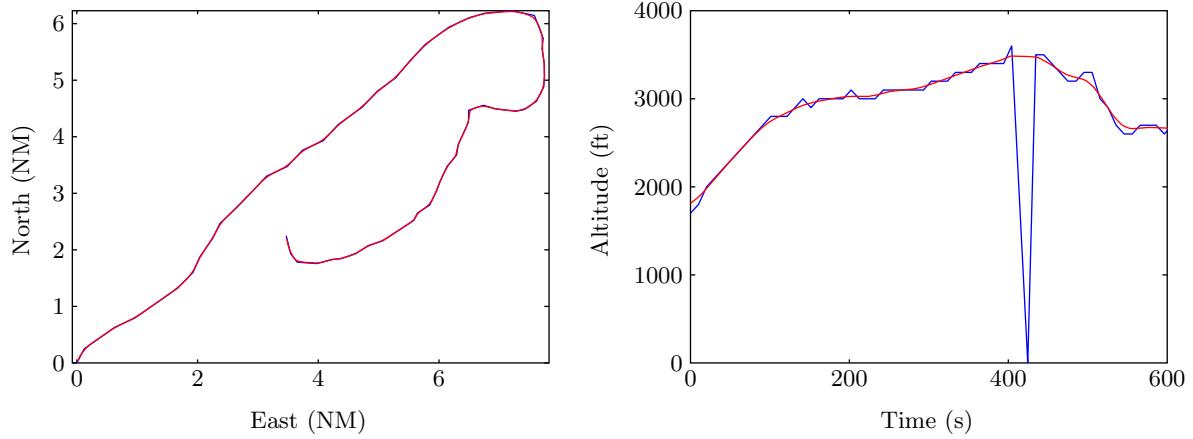
Figure 7. Preprocessing. Blue lines show an example raw track. Red lines show the track after outlier removal, smoothing, and interpolation.
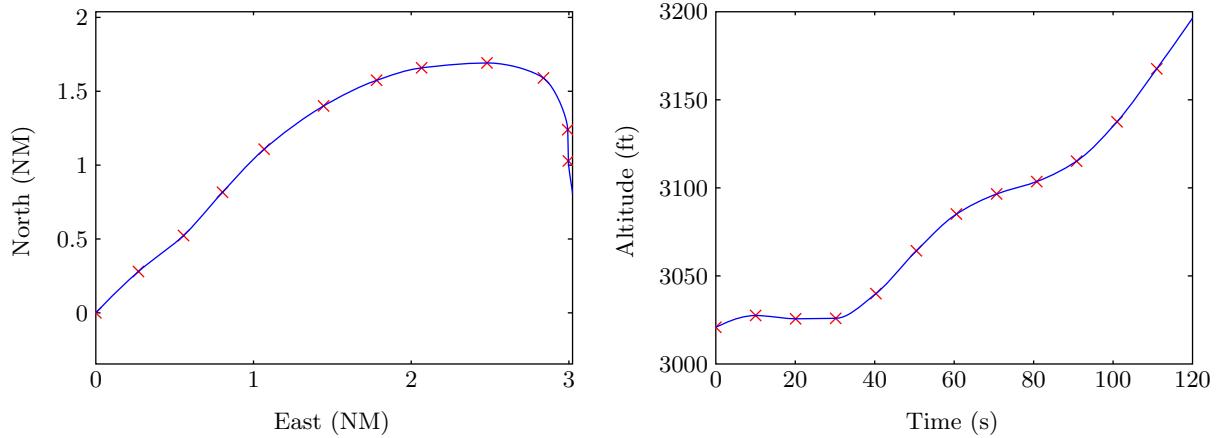
Figure 8. Piecewise-cubic Hermite interpolation on an example smoothed track. Red crosses indicate smoothed data points, and the blue curve shows the interpolation.

reports reduces the amount of data available to create the encounter model (usually undesirable), the amount of data discarded is relatively small, and hence this is a reasonable trade.

One motivation for the tails filter is the observation that smoothing during preprocessing of tracks can introduce distortions near their start and end. To illustrate these edge effects, Figure 9 provides the distribution of a sample of radar reports by displacement due to smoothing, which is the lateral distance between each raw report and its smoothed value. The distributions for edge points, or those within 60 s of the start or end of a track, and middle points—all other points—are plotted separately in addition to the full set of points. Edge points represent less than 20% of the total. The figure shows that edge points tend to be displaced farther from their raw values than middle points do.

The choice of the amount of data to remove must consider details of the smoothing method. As discussed in Section 3.4, smoothing of position data in source tracks is done using a local

20
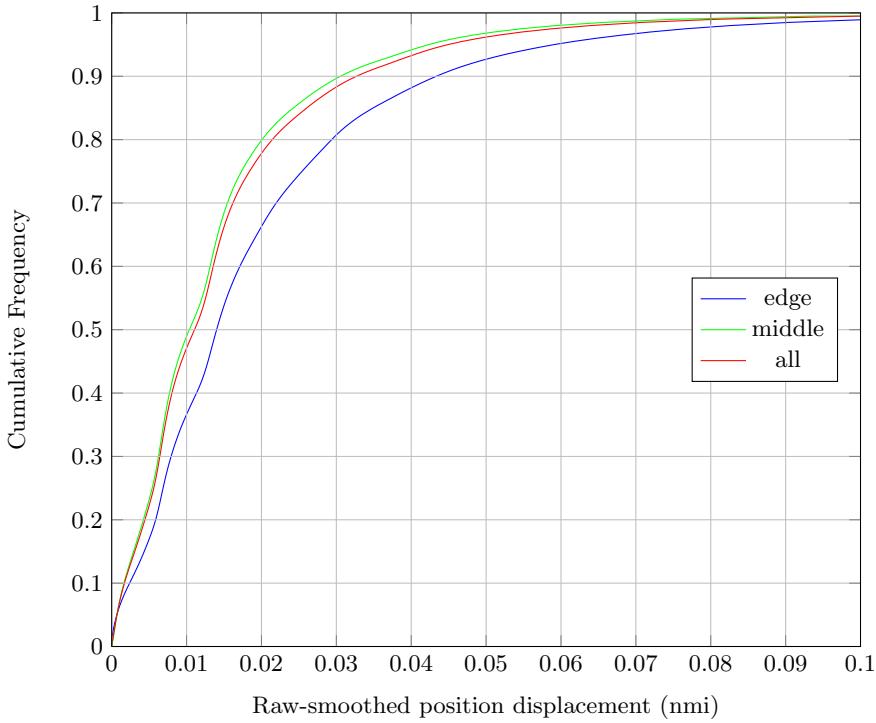
Figure 9. Cumulative distribution of radar reports by displacement due to smoothing; before the tails filter, edge points tend to be displaced farther from their raw values than middle points do.

Gaussian (normal) kernel with a standard deviation of 15 s vertically and 5 s horizontally. A normal distribution drops to about 1% of its maximum at 3 standard deviations from the mean and below 0.1% of its maximum at 4 standard deviations; consequently, since 60 s represents at least 4 standard deviations for both smoothing kernels, edge effects of smoothing should be mostly confined to the first and last 60 s of a track.

Another motivation for the tails filter is the difference between raw and processed airspeeds near the start and end of each track. This processing error is a consequence of the interpolation and occurs for all airspeeds and tracks. Figure 10 shows an example in which raw and processed position data are very similar but there is a noticeable airspeed difference at the tails of the track.

Any tracks with length 120 s or less are ignored entirely because they are shorter than the total amount of time removed. However, because the median source track length is more than 500 s, most tracks have reports remaining after this filter.

3.7   FEATURE EXTRACTION

Feature extraction involves converting an interpolated track into sequences of quantized bins representing geographic locations, airspace classes, altitude layers, airspeeds, vertical rates, turn rates, and accelerations.
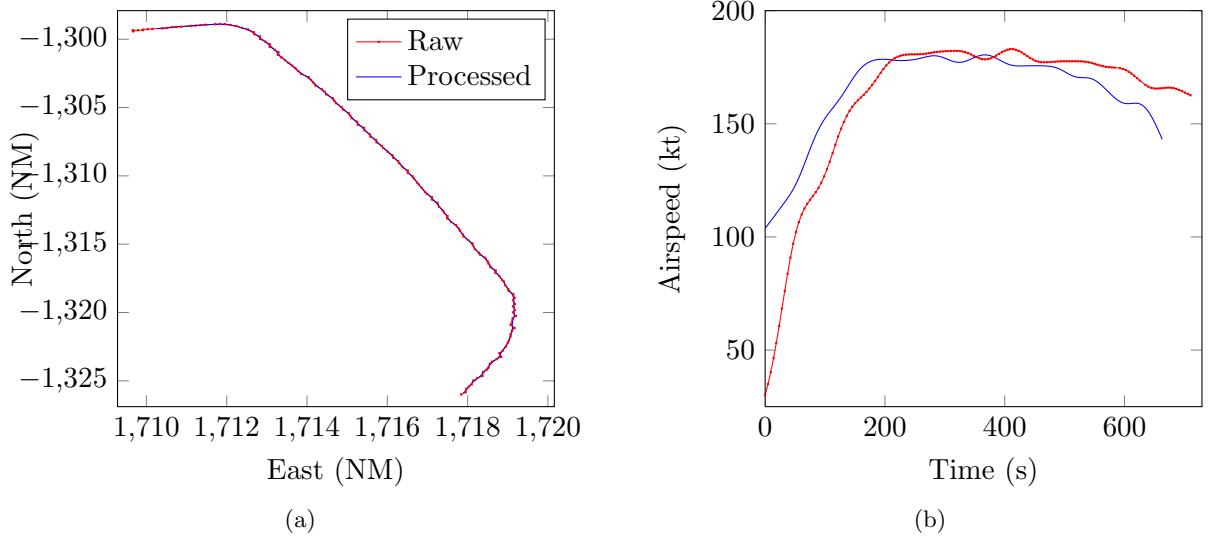
Figure 10. Velocity difference example in processing without tails filter; processing introduces airspeed differences near the start and end of this track.

- **Geographic location:** This variable is included in the model to account for variations in aircraft behavior based on location. It has four possible values corresponding to different geographic regions:

    1. Contiguous United States (CONUS), Alaska, Canada, and Mexico,
    2. Islands,
    3. CONUS Offshore, and
    4. Islands Offshore.

    These regions are defined by boundary polygons and are limited to the coverage areas of the radars used to build the model; for example, due to radar coverage limits, the CONUS region does not include all of Canada and Mexico, but only portions of these nations near U.S. borders (Figure 1). The islands offshore region represents aircraft operating in the Caribbean and Hawaii, while the mainland offshore region represents aircraft operating off the coasts of Alaska, the Gulf of Mexico, and the east and west coasts of CONUS. Geographic location is determined for each radar report individually, and hence tracks which cross region boundaries have reports assigned to different regions.

- **Airspace class:** An algorithm developed at Lincoln Laboratory estimates latitude and longitude of radar returns [22]. From estimates of altitude, latitude, and longitude, the airspace class is determined by searching through the National Airspace System Resources (NASR) database provided by the FAA. Since altitude estimates are based on Mode C reports of pressure altitude, uncorrected for barometric variation, it is possible that airspace class for some tracks is identified incorrectly. This limited inaccuracy in airspace class identification exists due to barometric variation and has a negligible impact on the model.

22

- **Altitude layer:** Altitude above ground level (AGL) determines the altitude layer in the model. By subtracting an estimate of ground elevation from pressure altitude, altitude AGL is estimated. The ground elevation estimates come from Digital Terrain Elevation Data (DTED) provided by the National Geospatial-Intelligence Agency (NGA). DTED Level 0 is used, which has post spacing of 30 arcseconds (approximately 900 meters).

- **Airspeed:** The true airspeed at time t is estimated by

$$v(t) = \sqrt{(x(t+1) - x(t))^2 + (y(t+1) - y(t))^2 + (h(t+1) - h(t))^2}\,.$$

- **Vertical rate:** The vertical rate is estimated from the smoothed and interpolated altitudes estimated from Mode C reports. The vertical rate at time t is given by $\dot{h}(t) = h(t+1) - h(t)$.

- **Turn rate:** First, compute the heading along the interpolated track. The heading at time t is given by $\psi(t)$ and corresponds to the direction from $(x(t), y(t))$ to $(x(t+1), y(t+1))$. To compute the turn rate at time t, find the acute change in heading between $\psi(t)$ and $\psi(t+1)$. Turns to the right have positive turn rates, and turns to the left have negative turn rates.

- **Acceleration:** To find the acceleration at a particular point, average the change in airspeed per unit time looking forward one time step and looking back one time step.

Next, the extracted features are smoothed using the same smoothing scheme used for tracks (Section 3.4). For turn rate, airspeed, and acceleration, set σ to 10 s, 20 s, and 20 s, respectively. These values are large enough so that noise is removed from the measurements but low enough so that the underlying properties of the maneuvers are not lost. Vertical rates are not smoothed in this step because the altitudes are already smoothed (Section 3.4).

In order to use a discrete Bayesian network, it is necessary to quantize the features by defining a sequence of cut points $c_1, \ldots, c_n$. Values less than c_1 are in the first bin, values greater than c_n are in the $(n+1)$th bin, and values in the half-open interval $[c_{i-1}, c_i)$ are in the ith bin. The cut points used for quantization are listed in Table 3. For example, referring to Table 3, all airspeed values less than 30 kt are placed into one bin; airspeeds between 30 kt and 60 kt are placed in the next bin, and so on. The cut points were chosen to capture the variation of the features as shown in the histograms in Figure 11.

Figure 12 shows the result of feature extraction on the same track shown in Figure 7.

3.8   STATISTICS EXTRACTION

With structures for the initial and transition distributions and the quantized features from a set of tracks, sufficient statistics are collected to estimate the model parameters. For the two Bayesian networks, the sufficient statistics are simply the counts N_{ijk} of the various features (see Appendix F). Appendix A describes the sufficient statistics extracted from the data.

The counts N_{ijk} are then compiled into a separate table for each variable. For example, the table for airspace class A (which is dependent on geographic location G) is shown in Table 4. Each

TABLE 3

Cut points used for feature quantization.

	Cut Points
G	CONUS, Islands, CONUS Offshore, Islands Offshore
A	B, C, D, O
L	$1200, 3000, 5000$
v	$30, 60, 90, 120, 140, 165, 250$
$\dot{v}$	$-1, -0.25, 0.25, 1$
$\dot{h}$	$-1250, -750, -250, 250, 750, 1250$
$\dot{\psi}$	$-6, -4.5, -1.5, 1.5, 4.5, 6$
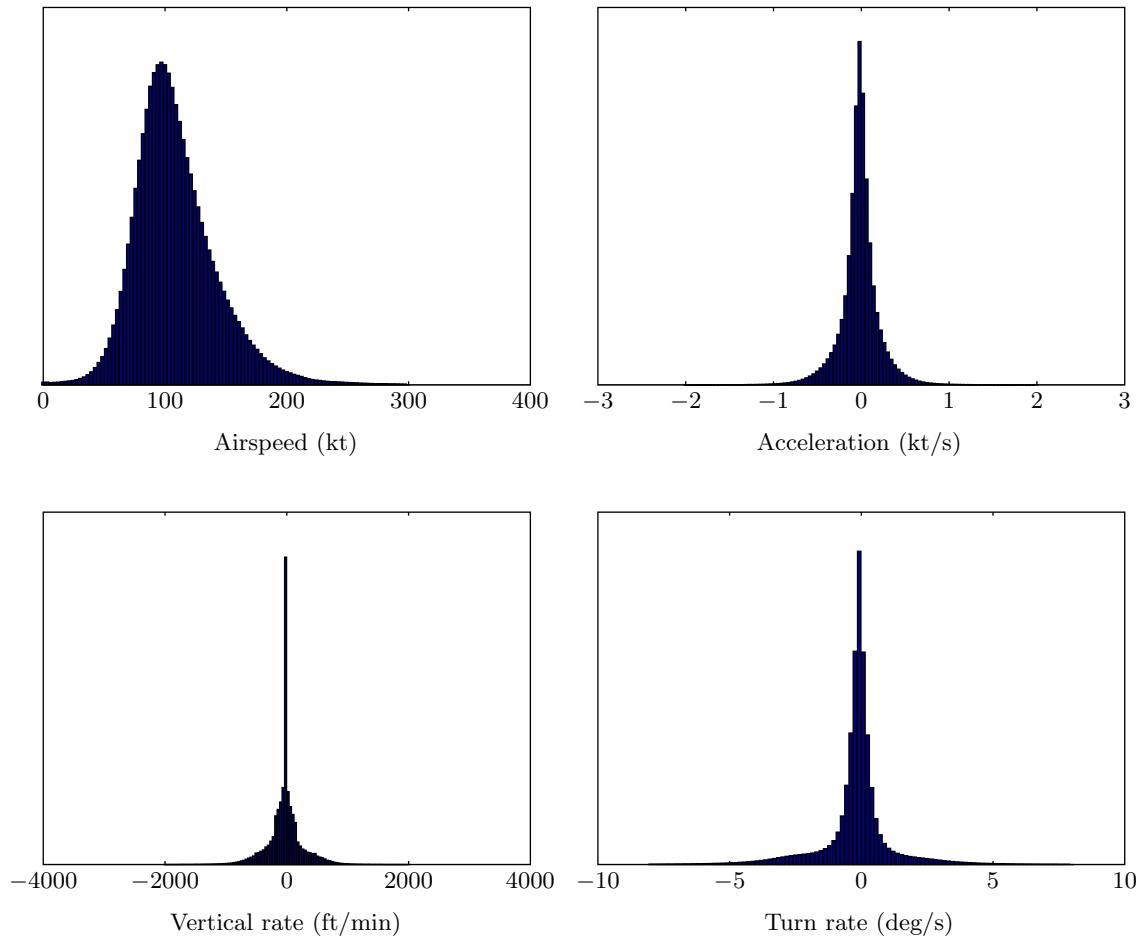
Figure 11. Feature histograms of recorded radar data based on 193 million samples.
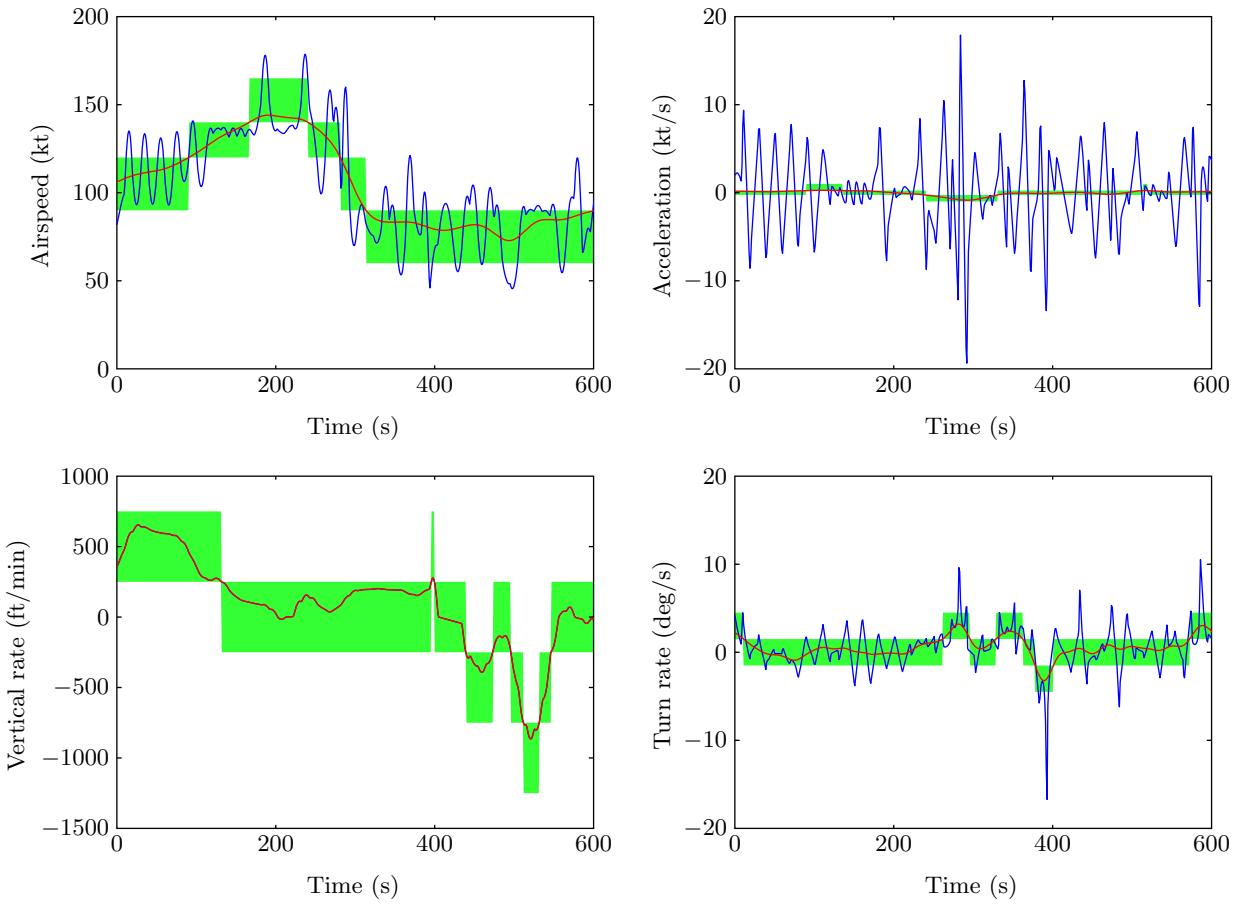
Figure 12. A plot of extracted features over time. Blue lines show features before smoothing, and red lines show features after smoothing. The green blocks show the bins to which the features are assigned.

25

cell entry in the table represents the counts N_{ijk} for that combination of parent bin G (1, 2, 3, 4) and bin of A (B, C, D, O). An electronic file available from Lincoln Laboratory contains all of the data required to construct these tables.

TABLE 4

$N(A \mid G)$

G	A			
	B	C	D	O
1	86714253	21392426	41858942	825467616
2	3703669	1361813	1548412	9911564
3	3689157	1027999	655019	47916919
4	2314332	462377	1337349	15379242

3.9   SUMMARY

To summarize, this section describes the process used to construct a model of uncorrelated aircraft trajectories based on radar data. Raw radar data were processed through multiple filtering and tracking stages and then used to populate a series of conditional probability tables organized within Bayesian networks. Each table cell represents the probability of a particular feature taking on a value within a certain quantized bin. Section 5 describes how to use these tables to generate random, but statistically representative, trajectories through sampling. An example of producing encounter trajectories using the model is given in Appendix B.

4.   STRUCTURE SEARCH

This section describes the process of seeking the best representation of the conditional independence relationships between variables. These relationships are represented by a set of directed edges, or connections, between variables, which can be visualized graphically as in Figure 3 or as a binary adjacency matrix (Appendix A) indicating which pairs of variables are connected. Each edge is directed from a parent variable to one of its child variables.

Not all structures represent these relationships equally well. Better structures allow the model to better represent the behavior of real aircraft and to produce more realistic synthetic tracks for use in fast-time simulations. However, since the number of possible structures is superexponential in the number of variables, exhaustive search of all possibilities is costly and may be infeasible for larger networks. Instead, the space of candidate structures for the model is prioritized and selectively sampled.

The process begins with a candidate structure motivated by previous encounter modeling results and by intuition about the relationships between variables. A preliminary version of the model with this structure is then generated using the steps in Section 3. A fully-connected starting structure is used because it allows greater flexibility when evaluating structures. Alternate models based on different candidate structures are then generated by removing one or more edges at a time. Structures are evaluated and compared using the Bayesian score as a metric (Appendix F). After a best structure is identified, the model is recreated to match it.

Due to the large space of possible structures, exhaustive search is done only on a smaller set of structures which are most similar to the starting structure—such as those in Figure 13. Figure 13 contains examples of candidate structures for the initial network—each obtained by removing one edge from the fully-connected network in Figure 3a. Because these structures have few edges removed from the starting structure, they can be computed from the starting structure at relatively small computational cost.

As the number of simultaneous removals approaches half the size of the starting connection set, the cost of exhaustive search increases due to more removals per candidate structure and more candidate structures to check[5]. Therefore, this larger subset of structures is instead explored via random sampling.

The set of initial network structures considered contains only those with the chosen variable ordering described in Appendix A. The set of transition network structures considered is constrained by allowing only static variables (G, A, and L) and those at time t to be parents and only variables at time $t + 1$ to be children. These conditions limit each network to at most 21 parent-child connections. Since each of these 21 possible connections may be present or not, they form a space of $2^{21} = 2{,}097{,}152$ possible structures for each network.

---

[5]The number of candidate structures which can be generated by removing k connections from a starting set of n is equal to the binomial coefficient $\binom{n}{k}$; this expression is maximized for fixed n when k is close to n/2 and minimized when k is 0 or n.

Overall, about 400,000 (20%) of the 2^{21} possible initial structures and about 290,000 (14%) of the 2^{21} possible transition structures were checked during the search leading to the final structures chosen in Figure 3. These two structures are the best observed among all structures checked, though they are not guaranteed optimal. The chosen initial network structure matches the candidate structure used to start the search process; no set of edge removals was found to improve it. However, the chosen transition network structure does improve on the starting candidate; among other changes, it drops all connections with the geographic variable G, which suggests that aircraft maneuvering behavior is generally independent of the geographic location where an aircraft operates.

The set of structures searched for this model is expanded relative to the set searched for the uncorrelated encounter model version 1.0, which considered fewer than 20 hand-selected candidate structures for each network [1].

(a) $G \rightarrow A$ removed

(b) $G \rightarrow L$ removed

(c) $G \rightarrow v$ removed

(d) $G \rightarrow \dot{v}$ removed
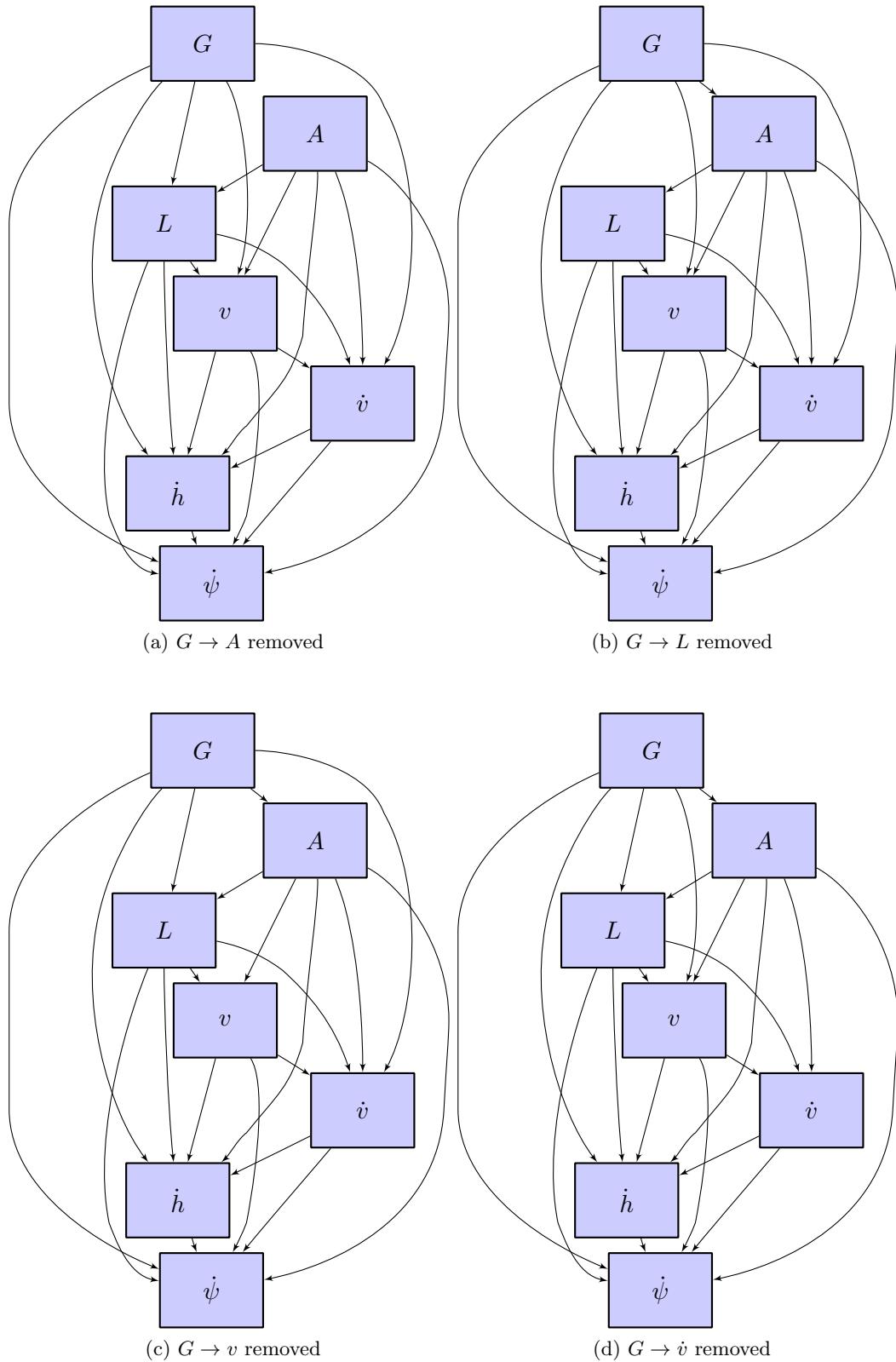
Figure 13. Example candidate initial networks.

This page intentionally left blank.

5.   SAMPLING

Once the data have been processed as described above, one can use the model structure and sufficient statistics to produce new trajectories that are representative of the ones observed by radar. The first step involves sampling from the discrete Bayesian network tables representing the initial and transition distributions. This provides a series of bins that represent coarse values to be used for each feature. The second step involves converting the coarse, discrete samples into fine, continuous samples by sampling within bins. Figure 14 illustrates this process.
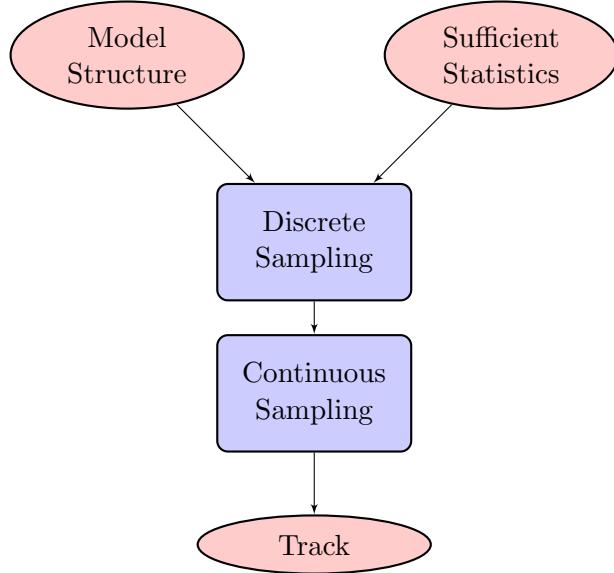
Figure 14. Sampling process flow.

5.1   DISCRETE SAMPLING

First, the Bayesian network representing the initial state distribution is sampled. Referring back to Figure 3a, first one would randomly sample from the table describing $N(G)$ to determine the bin to use for geographic location G. Given that bin for G, one would next sample from the table $N(A \mid G)$ to determine the bin for airspace class A and so on. See Appendix B for a more detailed description of this process.

Formally, consider sampling from a table describing the ith variable X_i (e.g., where $X_i =$ airspace class). As was shown in Table 4, each table contains a series of bins k, such as B, C, D, and O for airspace class, whose likelihood depends on the parent bin j, such as 1, 2, 3, and 4 for geographic location.

The probability assigned to bin k is then given by

$$\frac{\alpha_{ijk} + N_{ijk}}{\sum_{k'=1}^{r_i}(\alpha_{ijk'} + N_{ijk'})} \, , \tag{3}$$

where

- j is the instantiation of the parents of X_i in the Bayesian network,

- N_{ijk} is the number of times X_i was equal to k when its parents were instantiated to j in the data,

- α_{ijk} is a Dirichlet prior parameter, and

- r_i is the number of ways to instantiate X_i.

For this model, the prior is assumed to be $\alpha_{ijk} = 1$ because all combinations of relative frequencies for k are assumed equally probable. Sampling from the posterior distribution with $\alpha_{ijk} = 1$ is equivalent to adding 1 to all the counts in the tables in Appendix A and sampling according to the resulting relative frequencies; this assumption also ensures that there are no transitions with zero probability in the Markov model.

For instance, referring to Table 4, if G had previously been selected to be 1, then the probability of selecting airspace class O would be $P(A = O \mid G = 1) = (1 + 825467616)/(1 + 86714253 + 1 + 21392426 + 1 + 41858942 + 1 + 825467616) = 0.8463$.

The dynamic Bayesian network representing how the state changes can be sampled from the trajectory's initial state. The variables G, A, L, v, $\dot{h}(t)$, $\dot{\psi}(t)$, and $\dot{v}(t)$ are assigned using the standard Bayesian network sampling scheme to determine $\dot{h}(t+1)$, $\dot{\psi}(t+1)$, and $\dot{v}(t+1)$. The process may be repeated for the trajectory's intended duration.

The sample from the Bayesian network determines the bins for airspeed, vertical rate, turn rate, and acceleration. The bins are then sampled within as discussed in Section 5.2.

5.2 CONTINUOUS SAMPLING

To produce a continuous sample given a coarse, discrete sample from the initial distribution, a simple uniform sample within the bins is required. For example, if the initial airspeed is within the bin $[60, 90)$, a sample from the uniform distribution over the half-open interval $[60, 90)$ is required. Because the first and last bins associated with each interval are unbounded, it is necessary to impose some bounds. Table 5 shows the quantization boundaries based on the limits observed in the radar data.

With regard to the transition network, the question arises whether to resample continuous values when a variable remains in the same discrete bin over multiple time steps. In these cases, instead of sampling at every time step within bins for turn rate, vertical rate, and airspeed acceleration, a new continuous sample is produced with fixed, non-zero probability per time step;

TABLE 5

Sampling boundaries.

	Boundaries
h	$500, 1200, 3000, 5000, 12500$
v	$0, 30, 60, 90, 120, 140, 165, 250, 300$
$\dot{v}$	$-2, -1, -0.25, 0.25, 1, 2$
$\dot{h}$	$-2000, -1250, -750, -250, 250, 750, 1250, 2000$
$\dot{\psi}$	$-8, -6, -4.5, -1.5, 1.5, 4.5, 6, 8$

this probability has been estimated for each variable from the radar data. This strategy prevents numerous minor changes and allows relatively steady-state flight conditions to occur at the frequency observed in the radar data. The within-bin transition rates are estimated from the data by introducing three smaller bins within each bin and computing the relative rate that tracks stay within a single smaller bin versus moving to another small bin within the same coarse bin. The Eurocontrol cooperative encounter model used a similar strategy [6].

When producing continuous samples from bins that include zero in their range, the sampling process produces zero instead of sampling uniformly; this method prevents generation of unrealistically small vertical rates, turn rates, and accelerations.

Figure 15 plots an example of vertical rates, turn rates, accelerations, and airspeeds generated by sampling from the Bayesian networks. First, coarse bins are selected randomly and shown as green bars. Next, fine values within each bin are selected and shown in blue. Note that vertical rate, for example, is held to precisely zero when the bin spans zero; within non-zero bins it may be reselected several times at the mean rate described above. Figure 16 shows the resulting track produced by the sampled features shown in Figure 15. Translation of features into tracks involves running a discrete-time simulation as described in the next section.
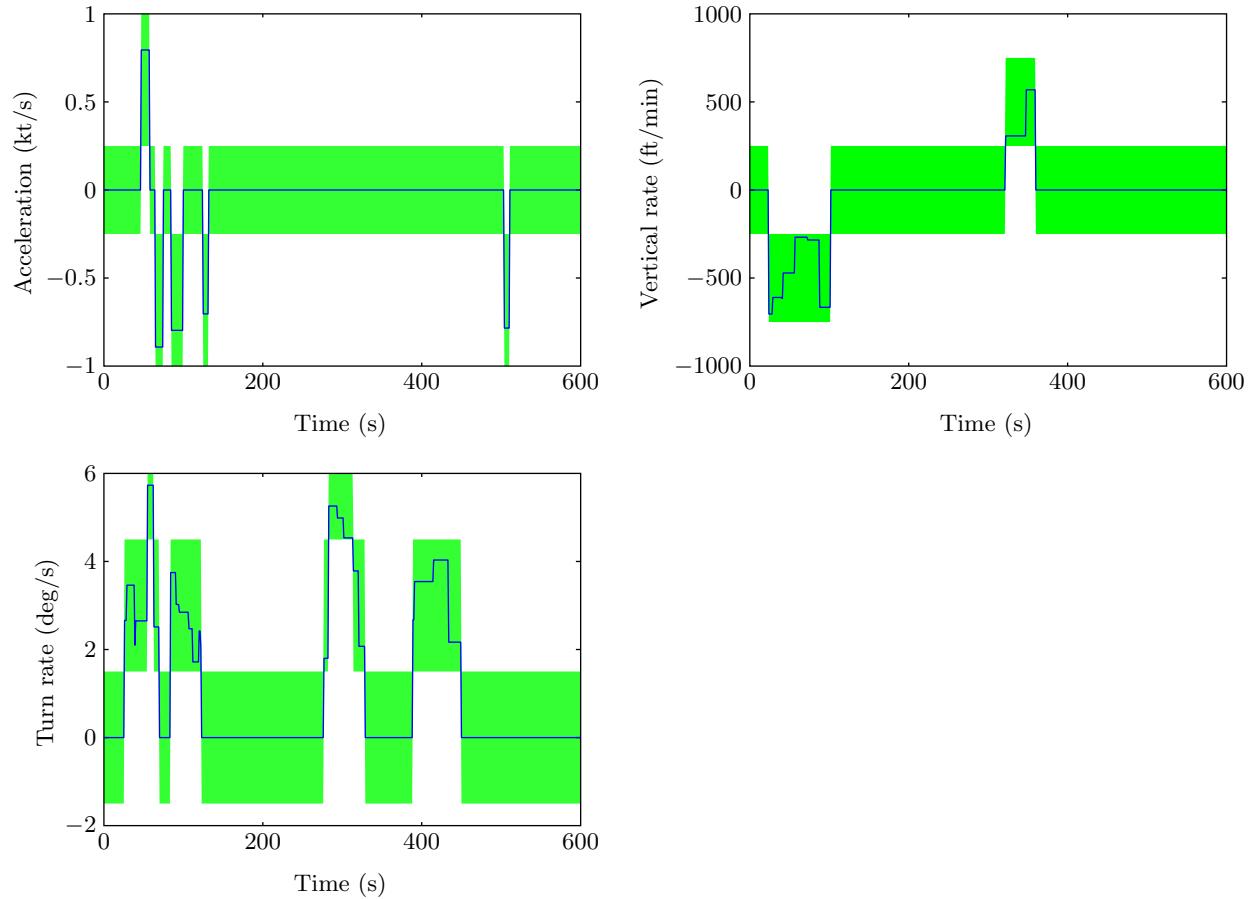
Figure 15. A plot of sampled features over time. Green blocks show the coarse discrete bins and blue lines show the resulting fine samples within those bins.
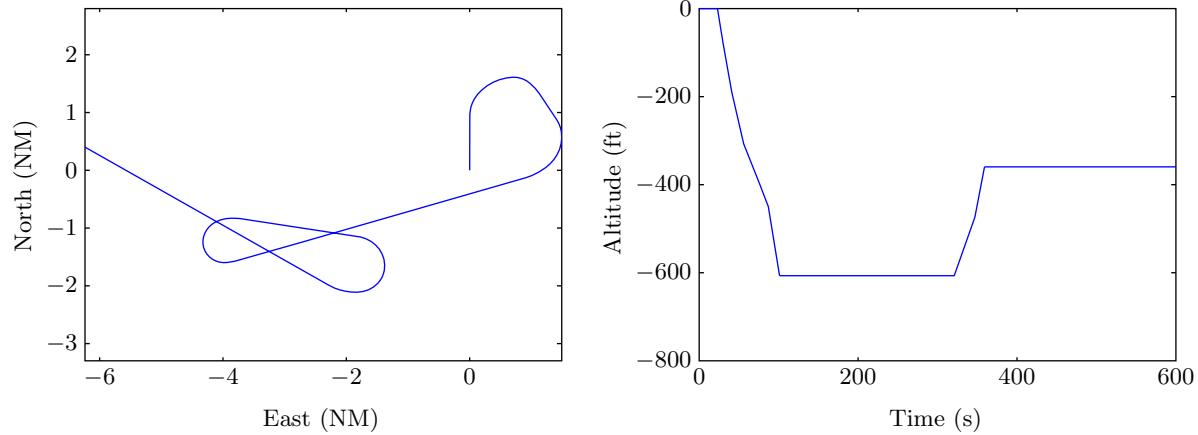

Figure 16. A track generated by sampling from the initial and transition distributions. Positions and altitudes are shown relative to the initial position and altitude.

6. SIMULATION

Earlier sections explained how to build a dynamic probabilistic model of aircraft and how to sample from the model to produce nominal trajectories representative of those observed in the radar data. The same modeling techniques discussed earlier may be used to build a model of the manned or unmanned aircraft with the avoidance system to be evaluated. This section explains how to combine a model of the aircraft with the avoidance system to be evaluated, AC1, with a model of an intruder, AC2, into a simulated encounter.

The trajectory for AC1 may be specified by the analyst (e.g., to focus on a particular phase of flight), based on actual flight paths from mission planning or radar data, or randomly generated using a statistical model representative of that aircraft's typical flight profiles. In a study of conventional VFR-on-VFR encounters, for example, trajectories for both AC1 and AC2 could be generated using the uncorrelated encounter model described here.

Simulated encounters that are extremely unlikely to result in near mid-air collisions (NMACs) are avoided by focusing computational effort on encounters that occur in an encounter cylinder centered on AC1. AC2 is initialized at a random location on the surface of the encounter cylinder and the dynamic models are used to update the states of AC1 and AC2 over time. If the avoidance system on AC1 commands an avoidance maneuver, it overrides the nominal rates suggested by the dynamic model. If AC2 enters a NMAC cylinder or exits the encounter cylinder, the encounter run is terminated. The NMAC cylinder has radius $r_{\mathrm{nmac}} = 500\,\mathrm{ft}$ and height $2h_{\mathrm{nmac}} = 100\,\mathrm{ft}$.

6.1 ENCOUNTER CYLINDER DIMENSIONS

The encounter cylinder has radius r_{enc} and height $2h_{\mathrm{enc}}$. The appropriate dimensions of the encounter cylinder depend on the aircraft dynamics and avoidance system. If the encounter cylinder is too small, the avoidance system will not have enough opportunity to be fully exercised before a collision. If the encounter cylinder is too large, then computation is wasted.

An upper bound for r_{enc} is the amount of time required by the avoidance system to detect, track, and avoid a target multiplied by the sum of the maximal airspeeds of AC1 and AC2. An upper bound for h_{enc} is the amount of time required by the avoidance system to detect, track, and avoid a target multiplied by the sum of the maximal vertical rate magnitudes of AC1 and AC2.

6.2 ENCOUNTER INITIALIZATION

Rejection sampling is used to generate the initial conditions of an encounter involving two aircraft: AC1 and AC2; it involves proposing a series of candidate samples from a random distribution until choosing one that meets a set of criteria. The process used for generating initial conditions for encounters is as follows:

1. Generate a set of initial conditions for AC1. The initial conditions include geographic location, airspace class, altitude layer, airspeed, vertical rate, turn rate, and acceleration.

2. Sample from the Bayesian network to generate AC2. If the sample has the same altitude layer and geographic location, then keep AC2; otherwise, reject the sample and continue to sample until the airspace class, altitude layer, and geographic location match AC1[6].

3. Randomly select the heading of AC1 and AC2 over the interval $[0, 360]$ degrees.

4. Calculate the velocity vectors for both aircraft ($\boldsymbol{v}_1$ and $\boldsymbol{v}_2$) based on the aircraft initial conditions. The relative velocity vector $\boldsymbol{v}_r$ is $\boldsymbol{v}_1$ subtracted from $\boldsymbol{v_2}$. See Figure 17.

5. Determine the projected surface $\mathcal{S}$ of the cylinder onto a plane that is perpendicular to $\boldsymbol{v}_r$.

6. The intruder aircraft penetrates the encounter cylinder uniformly over $\mathcal{S}$. Therefore, uniformly select a random point p inside $\mathcal{S}$.

7. Project this point back onto the encounter cylinder. There will be two candidate points. Select the one such that the intruder aircraft is penetrating the encounter cylinder. The following tests can be used to determine which candidate initial point is correct:

   - If AC2 was initialized on the top of the encounter cylinder, accept the sample if the vertical rate of AC2 relative to AC1, denoted $\boldsymbol{v}_{r,v}$, is negative. This ensures that AC2 is penetrating the encounter cylinder for the first time.

   - If AC2 was initialized on the bottom of the encounter cylinder, accept the sample if the vertical rate of AC2 relative to AC1, denoted $\boldsymbol{v}_{r,v}$, is positive. This ensures that AC2 is penetrating the encounter cylinder for the first time.

   - If AC2 was initialized on the side of the encounter cylinder, accept the sample if $\hat{\boldsymbol{R}}_h \cdot \boldsymbol{v}_{r,h}$ is negative. Here $\hat{\boldsymbol{R}}_h$ is the horizontal component of $\hat{\boldsymbol{R}}$, which is the relative position of AC2 from AC1, and $\boldsymbol{v}_{r,h}$ is $\boldsymbol{v}_{1,h}$ subtracted from $\boldsymbol{v}_{2,h}$. The vectors $\boldsymbol{v}_{1,h}$ and $\boldsymbol{v}_{2,h}$ are the horizontal velocities of AC1 and AC2 respectively. If $\hat{\boldsymbol{R}}_h \cdot \boldsymbol{v}_{r,h}$ is negative, then accept the encounter because AC2 is penetrating the cylinder about AC1.

6.3 TRAJECTORY CONSTRUCTION

Once the initial conditions are selected, the dynamic models of AC1 and AC2 are used to update their trajectories over time.

Given the initial state of an aircraft (including at a minimum: position, altitude, heading, climb rate, turn rate, and velocity) and its control variables ($\dot{h}$, $\dot{\psi}$, and $\dot{v}$) the aircraft state is updated in $\leq 1\,\text{s}$ time steps using a dynamic model. Due to the wide variety of possible dynamic model implementations [23], details for computing this state update are not provided here. A basic approach would be to apply simple point-mass kinematics to update the aircraft state without considering what the aircraft may actually be doing in terms of bank angle, pitch rate, etc. More complex implementations could include 6-degree-of-freedom dynamic models in which the control

---

[6]Note that this will result in an incorrect distribution over layer and geographic location (compared to that observed). Section 7.3 describes how to correct for this.
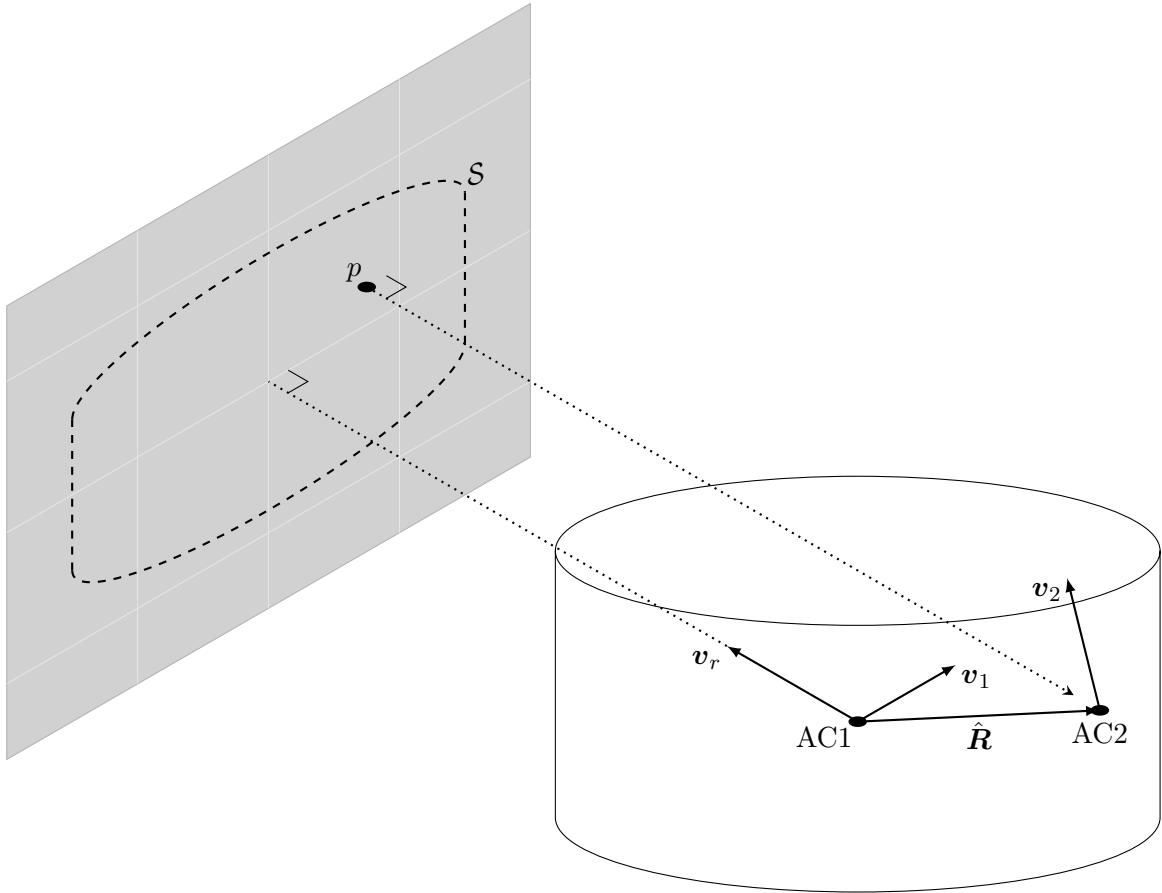
Figure 17. Initialization process.

variables are treated as target states provided to an autoflight control system which then applies the necessary control deflections to attain those targets. Lincoln Laboratory's Collision Avoidance System Safety Assessment Tool (CASSATT) typically uses a 4-degree-of-freedom model to update aircraft state by applying the necessary airspeed acceleration, roll rate, and pitch rate to achieve the target values for $\dot{h}$, $\dot{\psi}$, and $\dot{v}$ assuming curvilinear motion with a zero-sideslip constraint. The zero-sideslip constraint can be relaxed, for instance, if it is necessary to model transient dynamics with higher fidelity. Appendix D explains how to validate trajectories produced by other simulation software against trajectories produced by CASSATT.

A simulation run terminates when the intruder either exits the encounter cylinder or penetrates the NMAC cylinder. After running many simulations, $P(\text{nmac} \mid \text{enc})$ is roughly estimated by dividing the number of runs that resulted in NMAC by the total number of runs. Additional steps described in Section 7 can improve this estimate.

In addition to choosing the dynamic model, the analyst must also choose the length of tracks to sample from the model. Analysis in Appendix C shows that tracks sampled from the model are statistically valid only for a limited time interval of several minutes; sampled tracks longer than

this may be less representative of the modeled airspace. However, the use of longer tracks may be appropriate for some applications—this decision requires judgment by the analyst.

6.4 MULTIPLE ENCOUNTERS

The simulation currently only handles pairwise encounters. The probability of an intruder penetrating the encounter cylinder while another intruder is within the encounter cylinder is likely to be very small. One may compute this probability using

$$\int_0^\infty p(t)[1 - e^{\lambda_{\mathrm{enc}} t}]\mathrm{d}t = 1 - \int_0^\infty p(t) e^{\lambda_{\mathrm{enc}} t}\mathrm{d}t\,, \tag{4}$$

where $p(t)$ is the distribution over the amount of time intruders spend in the encounter cylinder and λ_{enc} is the rate at which new intruders penetrate the encounter cylinder. In the above equation, $1 - e^{\lambda_{\mathrm{enc}} t}$ comes from the cumulative density function for an exponential distribution. The possibility of simultaneous multiple intruders needs to be examined and will be an area of future work.

7. SAFETY EVALUATION

This section explains how to estimate the NMAC rate, denoted λ_{nmac}, based on a large number of simulations. First, observe that

$$\lambda_{\text{nmac}} = P(\text{nmac} \mid \text{enc})\lambda_{\text{enc}} \,,$$

where $P(\text{nmac} \mid \text{enc})$ is the probability that an aircraft that enters the encounter cylinder penetrates the NMAC cylinder before exiting the encounter cylinder and λ_{enc} is the rate at which aircraft penetrate the encounter cylinder. The mean time between NMACs is simply $\lambda_{\text{nmac}}^{-1}$.

This section makes the following assumptions:

1. the density of air traffic outside the encounter cylinder is uniform in the local region being studied[7], and

2. the trajectories of aircraft outside the encounter cylinder are independent of the trajectories of aircraft within the encounter cylinder.

From these two assumptions, $P(\text{nmac} \mid \text{enc})$ and λ_{enc} are computed. Note that estimating traffic density requires a more focused assessment of a particular region and time period which is beyond the scope of this report; this topic is discussed further in Section 7.2 and elsewhere [14]. However, sufficient radar data have been archived at Lincoln Laboratory to allow such an analysis for most regions of the nation, with the exception of certain areas affected by potential terrain masking.

Finally, this section also discusses correcting the estimate of $P(\text{nmac} \mid \text{enc})$ for altitude and geographic location, whose distributions may vary across different aircraft types and missions.

7.1 ESTIMATING NMAC PROBABILITY

Section 6 explained how to construct an encounter from two independent trajectories sampled from the distribution represented by Bayesian networks. By generating a large collection of encounters and determining which encounters lead to NMACs, one can estimate $P(\text{nmac} \mid \text{enc})$. Unfortunately, one cannot simply divide the number of sampled encounters that lead to NMACs by the total number of sampled encounters to estimate $P(\text{nmac} \mid \text{enc})$ due to the fact that the sampling scheme does not produce encounters from the same distribution that would occur in the airspace. In particular, the model generates encounters with aircraft velocities distributed identically to the aircraft population at large, despite the fact that in reality the distribution of aircraft velocities given that an encounter is occurring favors high-speed aircraft. Although one samples from a distribution that is different from the true distribution when constructing encounters, one can still use the samples to estimate $P(\text{nmac} \mid \text{enc})$ so long as one weights their results properly using an approach known as importance sampling [25]. This section begins by stating the weighting scheme and then proves that it is correct.

---

[7]More detailed studies relax this assumption [24].

Section 5 explained how to generate the trajectories for AC1 and AC2, z_1 and z_2, by sampling from the Bayesian networks with the requirement that both aircraft come from the same geographic location, airspace class and altitude layer. Section 6.2 then explained how to randomly select the position and orientation of AC2 relative to AC1, which is termed x_r. Importance sampling allows one to make the following approximation based on N samples

$$P(\text{nmac} \mid \text{enc}) \approx \frac{1}{N} \sum_i P(\text{nmac} \mid z_1^{(i)}, z_2^{(i)}, x_r^{(i)}, \text{enc}) \frac{V(z_1^{(i)}, z_2^{(i)})}{\bar{V}}.$$

The weight $V(z_1^{(i)}, z_2^{(i)})/\bar{V}$ corrects for the fact that the sampling distribution does not match the true distribution of encounter situations. The function $V(z_1^{(i)}, z_2^{(i)})$ is the average volume the encounter cylinder sweeps out per unit time when AC1 follows $z_1^{(i)}$ and the airspace consists exclusively of aircraft following $z_2^{(i)}$. In particular,

$$V(z_1^{(i)}, z_2^{(i)}) = |v_r| \, a(\mathcal{S}),$$

where $|v_r|$ is the magnitude of the relative velocity vectors and $a(\mathcal{S})$ is the area of $\mathcal{S}$. The constant $\bar{V}$ is the average volume the encounter cylinder sweeps out per unit time

$$\bar{V} = \iint p(z_1) p(z_2 \mid z_1) V(z_1, z_2) \, \mathrm{d}z_1 \mathrm{d}z_2.$$

Note that the distribution over z_2 is conditional on z_1 due to the constraint that AC1 and AC2 must belong to the same geographic location, airspace class and altitude layer. The constant $\bar{V}$ can be estimated using N samples:

$$\bar{V} \approx \frac{1}{N} \sum_i V(z_1^{(i)}, z_2^{(i)}). \tag{5}$$

Now that the weighting scheme has been defined, this section now proves that it is correct. From the laws of probability,

$$
\begin{aligned}
P(\text{nmac} \mid \text{enc}) \;=\;& \iint P(\text{nmac} \mid z_1, z_2, \text{enc}) p(z_1, z_2 \mid \text{enc}) \, \mathrm{d}z_1 \mathrm{d}z_2 \\
=\;& \iiint P(\text{nmac} \mid z_1, z_2, x_r, \text{enc}) p(x_r \mid z_1, z_2, \text{enc}) p(z_1, z_2 \mid \text{enc}) \, \mathrm{d}z_1 \mathrm{d}z_2 \mathrm{d}x_r.
\end{aligned}
$$

$P(\text{nmac} \mid \text{enc})$ may be approximated using Monte Carlo sampling. Since it is difficult to sample from $p(z_1, z_2 \mid \text{enc})$ directly, one can sample z_1 and z_2 from the distribution represented by the Bayesian network subject to the constraint that both aircraft come from the same geographic location, airspace class and altitude layer, and weight the samples appropriately:

$$P(\text{nmac} \mid \text{enc}) \approx \frac{1}{N} \sum_i P(\text{nmac} \mid z_1^{(i)}, z_2^{(i)}, x_r^{(i)}, \text{enc}) \frac{p(z_1^{(i)}, z_2^{(i)} \mid \text{enc})}{p(z_1^{(i)}) p(z_2^{(i)} \mid z_1^{(i)})}.$$

Next,

$$
\begin{aligned}
p(\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)} \mid \text{enc}) &= \frac{p(\boldsymbol{z}_1^{(i)})p(\boldsymbol{z}_2^{(i)} \mid \boldsymbol{z}_1^{(i)})}{\lambda_{\text{enc}}} \lambda_{\text{enc} \mid \boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}} \\
&\propto p(\boldsymbol{z}_1^{(i)})p(\boldsymbol{z}_2^{(i)} \mid \boldsymbol{z}_1^{(i)}) \lambda_{\text{enc} \mid \boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}} \\
&\propto p(\boldsymbol{z}_1^{(i)})p(\boldsymbol{z}_2^{(i)} \mid \boldsymbol{z}_1^{(i)}) V(\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}) \,.
\end{aligned}
$$

This result may be normalized to obtain

$$
\begin{aligned}
p(\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)} \mid \text{enc}) &= p(\boldsymbol{z}_1^{(i)})p(\boldsymbol{z}_2^{(i)} \mid \boldsymbol{z}_1^{(i)}) V(\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}) / \iint p(\boldsymbol{z}_1)p(\boldsymbol{z}_2 \mid \boldsymbol{z}_1) V(\boldsymbol{z}_1, \boldsymbol{z}_2)\, \mathrm{d}\boldsymbol{z}_1 \mathrm{d}\boldsymbol{z}_2 \\
&= p(\boldsymbol{z}_1^{(i)})p(\boldsymbol{z}_2^{(i)} \mid \boldsymbol{z}_1^{(i)}) V(\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}) / \bar{V} \,.
\end{aligned}
$$

Substitution and simplification leads to

$$
\begin{aligned}
P(\text{nmac} \mid \text{enc}) &\approx \frac{1}{N} \sum_i P(\text{nmac} \mid \boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}, \boldsymbol{x}_r^{(i)}, \text{enc}) \frac{p(\boldsymbol{z}_1^{(i)})p(\boldsymbol{z}_2^{(i)} \mid \boldsymbol{z}_1^{(i)}) V(\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}) / \bar{V}}{p(\boldsymbol{z}_1^{(i)})p(\boldsymbol{z}_2^{(i)} \mid \boldsymbol{z}_1^{(i)})} \\
&\approx \frac{1}{N} \sum_i P(\text{nmac} \mid \boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}, \boldsymbol{x}_r^{(i)}, \text{enc}) \frac{V(\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)})}{\bar{V}} \,,
\end{aligned}
$$

which corresponds to the weighting scheme defined above.

7.2 ESTIMATING ENCOUNTER RATE

Estimating the encounter rate requires knowing the density of traffic outside the encounter cylinder. This density, ρ, can be expressed in aircraft per NM^3. The rate at which aircraft enter the encounter cylinder is the product of ρ and the average volume of new airspace the encounter cylinder sweeps through per unit time, $\bar{V}$, which was discussed in Section 7.1:

$$
\lambda_{\text{enc}} = \rho \bar{V} \,. \tag{6}
$$

Values for ρ for VFR (1200-code) traffic over the continental United States can be estimated from radar data. However, it is important to note that density during the day is significantly higher than at night. Hence, if one is interested in estimating collision risk due to VFR aircraft when flying a particular unmanned aircraft during the day, for example, then one must use a value for ρ that is specific to the expected hours of operation so that collision risk is not underestimated.

The $\bar{V}$ in Equation 6 depends upon the size of the encounter cylinder and on the average velocities of aircraft involved in encounters. As one example, an estimate of $\bar{V}$ was obtained empirically by generating 1 million encounters from the uncorrelated encounter model and applying the approximation in Equation 5. If the encounter cylinder has radius 5 NM and height 3000 ft, then $\bar{V}$ was found to be approximately 937 NM^3/hr for trajectories generated from the uncorrelated encounter model. The average VFR traffic encounter rate for Miami-Dade County, for example,

is then $0.003 \times 937 = 2.8$ encounters per flight hour. Note that this does not imply that collision avoidance maneuvering would necessarily occur 2.8 times per hour because not all encounters require maneuvering. Most of these encounters dissipate benignly, but it is important to simulate them to ensure the avoidance system does not induce problems. Further, because this example density estimate was based on an average over two weeks, night and day, at all altitudes between 500 to 18,000 ft, it is important to reiterate the need to obtain density values for specific operating altitudes, airspaces, and times to get a more accurate estimate of the encounter rate for a particular operational concept. Finally, encounter rates for cooperative traffic and for noncooperative unconventional aircraft also need to be considered to obtain an overall collision risk estimate [24].

λ_{nmac} computes as follows:

$$
\begin{aligned}
\lambda_{\mathrm{nmac}} &= P(\mathrm{nmac} \mid \mathrm{enc})\lambda_{\mathrm{enc}} \\
&= \rho \bar{V} P(\mathrm{nmac} \mid \mathrm{enc}) \\
&\approx \frac{\rho \bar{V}}{N} \sum_i P(\mathrm{nmac} \mid \boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}, \psi^{(i)}, \boldsymbol{x}_r^{(i)}, \mathrm{enc}) \frac{V(\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)})}{\bar{V}} \\
&= \frac{\rho}{N} \sum_i P(\mathrm{nmac} \mid \boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}, \psi^{(i)}, \boldsymbol{x}_r^{(i)}, \mathrm{enc}) V(\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}).
\end{aligned}
$$

7.3 CORRECTING FOR LAYER AND GEOGRAPHIC LOCATION

Encounters are generated across altitude layers and geographic locations according to their distributions defined in the model. These distributions represent the observed rate of occurrence of aircraft in each altitude layer and geographic location. However, the expected encounter rate is proportional to the airspace density, *not* the cumulative occurrence of aircraft, in the local airspace. Furthermore, the aircraft of interest may be exposed to distributions of encounters that differ from the distributions in the model because the aircraft of interest has a greater exposure time (t_e) to encounters in certain altitude layer and geographic location combinations. Consequently, $P(\mathrm{nmac} \mid \mathrm{enc})$ for a particular aircraft of interest may differ from its average value across all observed situations. The adjusted mean NMAC probability over all geographic locations and altitude layers which considers these factors a posteriori is

$$
\bar{P}(\mathrm{nmac} \mid \mathrm{enc}) = \sum_i P(\mathrm{nmac} \mid \mathrm{enc}, l_i, g_i) P(l_i, g_i \mid \mathrm{enc}) = \frac{\sum_i P(\mathrm{nmac} \mid \mathrm{enc}, l_i, g_i) \rho_i \bar{V}_i t_e^{(i)}}{\sum_i \rho_i \bar{V}_i t_e^{(i)}}, \quad (7)
$$

where i denotes each altitude layer and geographic location combination. The term $P(l_i, g_i \mid \mathrm{enc})$ is the proportion of encounters expected in each altitude layer and geographic location combination.

If one knows ρ_i, $\bar{V}_i$, and $t_e^{(i)}$ a priori, then the altitude and airspace class distributions can be modified to reflect this knowledge before sampling from the model. Then, the mean NMAC probability estimate is simply the NMAC probability for all encounters. This sampling procedure may be useful when the aircraft of interest is expected to operate in a specific geographic location or altitude layer. If little is known about the expected operating environment, then an objective (i.e., uniform) assumption regarding the geographic location and altitude layer may be suitable.

8. SUMMARY

This report presents updates to a previous Lincoln Laboratory uncorrelated aircraft encounter model to enhance its scope and data processing methods. The updates are designed to increase the model's applicability to self-separation by permitting modeling of longer encounters lasting approximately 60 to 300 seconds. Additional processing steps seek to remove previously unaddressed anomalies both present in the historical data and introduced during processing. The new steps include discarding small portions of the historical radar data and removing additional types of outliers.

The updated model also includes an additional discrete variable specifying the geographic location, which allows a single model to provide specialized coverage of different geographic regions.

Like the previous Lincoln Laboratory uncorrelated model, the dynamics of aircraft state are modeled using a Markov approach where the probability of the next state depends only upon the current state. One way to represent a Markov model is with an exhaustive state-transition matrix that specifies the probability of transitioning between all pairs of states. However, the number of independent parameters required to define the matrix grows super-exponentially with the number of variables defining the model. The more independent parameters there are in the model, the more data one needs to properly estimate their values. However, using dynamic Bayesian networks and leveraging conditional independence between some variables greatly reduces the number of parameters. The dynamic Bayesian network structure is learned by maximizing the posterior probability of the network structure given the data.

The model presented in this report assumes the trajectories of the aircraft involved in an encounter are independent of each other prior to intervention by an avoidance system, human or automated. It assumes that aircraft blunder into close proximity without prior intervention.

This page intentionally left blank.

APPENDIX A
MODEL PARAMETERS

This appendix describes the sufficient statistics, N_{ijk} (see Appendix F), used to estimate the conditional probabilities associated with the initial and transition distributions. These sufficient statistics are based on beacon reports associated with aircraft squawking VFR (Mode A code 1200) from 3–9 March 2010, 3–9 June 2010, 3–9 September 2010, and 3–9 December 2010, which amounts to over 295,000 flight hours from across the United States after all processing steps—including track fusion, smoothing, interpolation, and filtering. Other parameters relevant to generating new encounters from the model are also described in this section.

A text file, available electronically from Lincoln Laboratory, describes the following model parameters:

- **Variable labels:** A quoted, comma-delimited list specifies the variable labels, e.g., `\dot \psi`, as would be used by LATEX. There are different variable labels for the initial network and the transition network. The ordering of the variables in this list determines the ordering of the variables in the other tables. Note that the ordering of the variable labels does not necessarily correspond to the order in which they are sampled; a topological sort may be necessary before sampling.

- **Graphical structure:** A binary matrix is used to represent graphical structure. A `1` in the ith row and jth column means that there is a directed edge from the ith variable to the jth variable in the Bayesian network; the ordering of the variables are as defined in the variable labels section of the file. The text file specifies two graphical structures: one for the initial network and the other for the transition network. The element in the ith row and jth column is represented by $G(i,j)$.

- **Variable instantiations:** For each network, a list of integers specifies the number of instantiations that exist for each variable.

- **Sufficient statistics:** For each network, a list of integers specifies the sufficient statistics; this appendix explains how to interpret it.

- **Boundaries:** The boundaries of the variable bins are specified by a row of numbers. The variables G, A, and L are not quantized because they are already discrete, and so boundaries do not exist. A $*$ is used for these variables.

- **Resampling rates:** A list of numbers specifies the resampling rates (Section 5.2).

The list of numbers describing the sufficient statistics, N_{ijk}, requires explanation. The array is ordered first by increasing k, then increasing j, and then increasing i. Again, the variable ordering is as defined in the variable labels section of the file. One way to load the sufficient statistics into memory is to allocate an array of pointers to 2-dimensional matrices. There would be 7 matrices for the initial network and 10 matrices for the transition network. The dimensions of each matrix is $r_i \times q_i$, or the number of instantiations of the variable by the number of instantiations of the

parents of the variable (see Appendix F)[8]. The counts may be read directly into the matrices from the file, starting with the first column of the first variable to the last column of the last variable.

Instead of reading the sufficient statistics into an array of matrices stored in memory, one can reference the elements in the parameters file directly. For some specified variable i, parental instantiation j, and variable instantiation k, the value N_{ijk} is given by the following element in the list

$$k + r_i(j-1) + \sum_{i'=1}^{i-1} q_{i'} r_{i'} \,, \tag{A-1}$$

where q and r are as specified in Appendix F[9].

It is important to clarify the ordering of the parental instantiations. If the variables $X_1, \ldots, X_n$ are instantiated to bins $b_1, \ldots, b_n$, the parental instantiation of variable X_i is given by

$$j = 1 + \sum_{i'=1}^{n} G(i', i)(b_{i'} - 1) \prod_{i''=1}^{i'-1} r_{i''}^{G(i'',i)} \,. \tag{A-2}$$

For example, suppose that a variable has three parents. The first parental instantiation will assign all parents to their first bin. The second parental instantiation will assign the first parent (as defined by the ordering in the variable labels portion of the file) to its second bin and the other two parents to their first bin. The sequence continues until all of the parents are instantiated to their last bins.

The following is a fragment of the parameter file. The lines that describe the sufficient statistics are truncated due to length.

```
# labels_initial
"G", "A", "L", "v", "\dot v", "\dot h", "\dot \psi"
# G_initial
0    1    1    1    1    1    1
0    0    1    1    1    1    1
0    0    0    1    1    1    1
0    0    0    0    1    1    1
0    0    0    0    0    1    1
0    0    0    0    0    0    1
0    0    0    0    0    0    0
# r_initial
4 4 4 8 5 7 7
# N_initial
975433237    16525458    53289094    19493300    86714253    21392426    41858942    ...
# labels_transition
```

---

[8]For the transition network, note that the matrices for the variables that are not associated with time $t+1$ are empty.

[9]In the transition network, q_i is as defined in Appendix F for the nodes representing variables at time $t+1$. For the other nodes (static variables and the variables at time t), q_i is set to zero because these nodes have no parents.

```
"G", "A", "L", "v", "\dot v(t)", "\dot h(t)", "\dot \psi(t)", "\dot v(t+1)",
"\dot h(t+1)", "\dot \psi(t+1)"
# G_transition
0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    1    1
0    0    0    0    0    0    0    1    1    1
0    0    0    0    0    0    0    1    1    1
0    0    0    0    0    0    0    1    1    1
0    0    0    0    0    0    0    1    1    0
0    0    0    0    0    0    0    1    0    1
0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0
# r_transition
4    4    4    8    5    7    7    5    7    7
# N_transition
5    0    0    0    0    0    0    0    0    0    0    0    0    ...
# boundaries
*
*
*
0 30 60 90 120 140 165 250 300
-2 -1 -0.25 0.25 1 2
-2000 -1250 -750 -250 250 750 1250 2000
-8 -6 -4.5 -1.5 1.5 4.5 6 8
```

This page intentionally left blank.

APPENDIX B
TRAJECTORY GENERATION

This appendix explains how to generate an encounter from the model. Software for parsing the parameters file (Appendix A) and generating trajectories is available from the authors.

B.1 INITIAL NETWORK SAMPLING

The first step in generating a random trajectory using the model is to sample from the Bayesian network representing the initial state distribution. To sample from a Bayesian network, as explained in Appendix F, one must first produce a topological sort of the nodes in the network. A topological sort orders the nodes of the network so that parents precede their descendants. The following is the graphical structure of the initial network as specified in the parameters file (see Appendix A) and shown in Figure B-1:

```
0    1    1    1    1    1    1
0    0    1    1    1    1    1
0    0    0    1    1    1    1
0    0    0    0    1    1    1
0    0    0    0    0    1    1
0    0    0    0    0    0    1
0    0    0    0    0    0    0
```

As can be seen, this ordering of the nodes is already topologically sorted: the first node (geographic location) is connected to all other nodes. The second node (airspace class) is connected to all following nodes, and so on. The final node ($\dot{\psi}$) is not connected to any other nodes.

With the nodes topologically sorted, begin by sampling the first variable. As specified in the parameters file, the first variable is G, geographic location. Equation F-5, reproduced below, shows how to produce a random sample:

$$P(X_i = k \mid \pi_{ij}, D, G) = \frac{\alpha_{ijk} + N_{ijk}}{\sum_{k'=1}^{r_i}(\alpha_{ijk'} + N_{ijk'})} \, .$$

In other words, the probability of selecting a particular geographic location is proportional to the prior (α_{ijk}) plus the frequency that geographic location appeared in the data (N_{ijk}). Following Cooper and Herskovits [26], use an objective prior and set α_{ijk} to 1. To determine the values for N_{ijk}, look at the sufficient statistics recorded in the parameters file. The sufficient statistics for the initial network is recorded as a long series of numbers. Use the method described in Appendix A to determine the actual values. These values turn out to be the first four numbers in the sufficient statistics sequence and are shown in Table B-1.
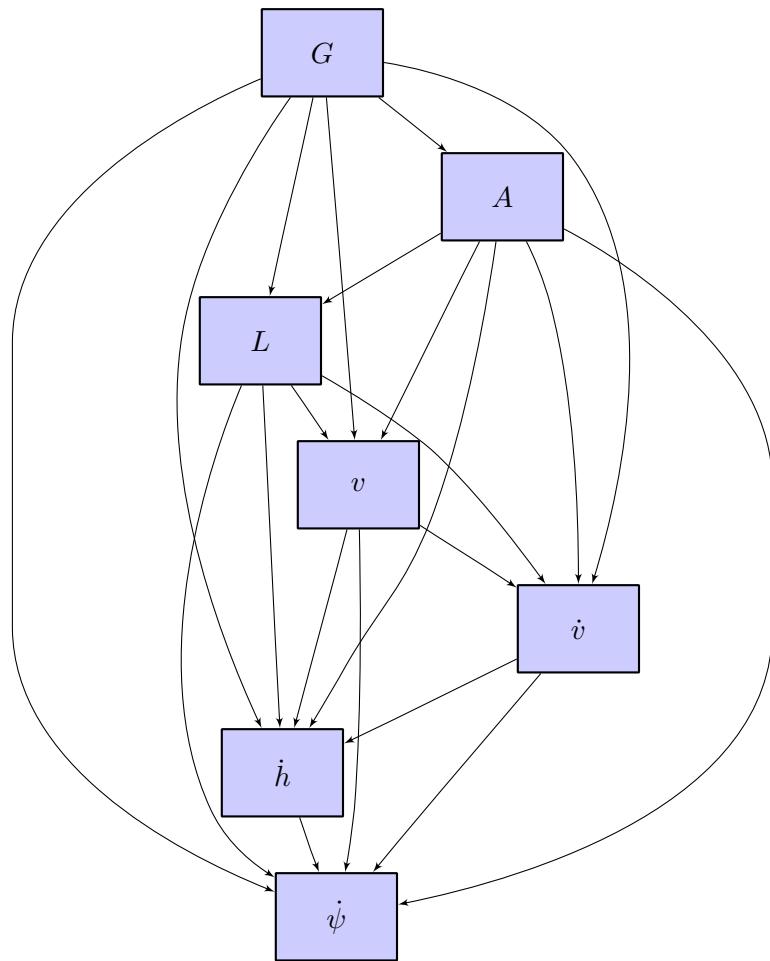
Figure B-1. A graphical representation of the initial network.

TABLE B-1

Sufficient statistics for geographic location, $N(G)$.

		G		
	1	2	3	4
	975433237	16525458	53289094	19493300

Now, compute the probability of selecting each value of G using Equation F-5:

- $P(G = 1) = (975433237 + 1)/(975433237 + 1 + 16525458 + 1 + 53289094 + 1 + 19493300 + 1) = 0.9161$

- $P(G = 2) = (16525458 + 1)/(975433237 + 1 + 16525458 + 1 + 53289094 + 1 + 19493300 + 1) = 0.0155$

- $P(G = 3) = (53289094 + 1)/(975433237 + 1 + 16525458 + 1 + 53289094 + 1 + 19493300 + 1) = 0.0500$

- $P(G = 4) = (19493300 + 1)/(975433237 + 1 + 16525458 + 1 + 53289094 + 1 + 19493300 + 1) = 0.0183$

Use a random number generator to choose a geographic location. For this example, suppose 1 is chosen—the first instantiation of G.

The next step is to instantiate the second variable, A, which is airspace class. Choosing a random instantiation for A requires an extra step because A depends upon other variables, namely G; this step requires computing the conditional probability distribution $P(A \mid G = 1)$, which is the distribution over the values of A given that G is 1. Consult the sufficient statistics table, $N(A \mid G)$, which is extracted from the parameters file (using the process in Appendix A) and displayed in Table B-2.

TABLE B-2

Sufficient statistics for airspace class given geographic location, $N(A \mid G)$.

		A		
G	B	C	D	O
1	86714253	21392426	41858942	825467616
2	3703669	1361813	1548412	9911564
3	3689157	1027999	655019	47916919
4	2314332	462377	1337349	15379242

Because only the counts associated with $G = 1$ are needed, consider only the first row of the table. To translate the counts into probabilities, add 1 to each element in the first row and then divide by the total resulting sum of that row. Again, use a random number generator to select an airspace class according to these probabilities. Suppose O is chosen.

The next variable to be assigned is L. Consult the table for L, focusing on the row (highlighted in Table B-3) that is consistent with the variable assignments so far: $G = 1$ and $A = O$. Choose a random altitude layer based on the counts as with the previous variables.

TABLE B-3

Sufficient statistics for altitude layer given geographic location and airspace class,
$$N(L \mid G, A).$$

		L			
G	A	[500, 1200)	[1200, 3000)	[3000, 5000)	[5000, ∞)
1	B	5568345	23858106	28152575	29135227
2	B	339649	1718413	1349929	295678
3	B	56476	662301	1222079	1748301
4	B	61839	744969	870960	636564
1	C	3489379	14185036	3677129	40882
2	C	198213	999163	164284	153
3	C	41514	692785	275594	18106
4	C	56986	317671	87320	400
1	D	22920532	17998283	299001	641126
2	D	974138	568319	4124	1831
3	D	226456	339305	6624	82634
4	D	232724	714072	294734	95819
1	O	99307858	308711233	147418646	270029879
2	O	2264236	4519902	1329154	1798272
3	O	3384105	11693898	7571628	25267288
4	O	1360754	4200866	1970933	7846689

This process of randomly assigning values to each variable conditioned on the values of its parents continues until all variables have been assigned. For this example, assume that the following assignments have been made:

- $G = 1$

- $A = O$

- $L = [500, 1200)$

- $v = [120, 140)$

- $\dot{v} = [-0.25, 0.25)$

- $\dot{h} = [-2000, -1250)$

- $\dot{\psi} = [-1.5, 1.5)$

Specific values within each bin are then determined based on a different uniform random number for each. In the case where a bin spans 0 (as it does in this example for $\dot{v}$, and $\dot{\psi}$), the value 0 itself is assigned.

B.2 TRANSITION NETWORK SAMPLING

The previous section described how to sample from the initial Bayesian network to generate a random initial state. The next step is to use the dynamic Bayesian network representing the transition distribution to generate the state at the next time step based on the initial state. The parameters file defines the following sequence of variables in the dynamic Bayesian network: $G, A, L, v, \dot{v}(t), \dot{h}(t), \dot{\psi}(t), \dot{v}(t+1), \dot{h}(t+1), \dot{\psi}(t+1)$. The file also defines the following graphical structure for the network:

```
0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    1    1
0    0    0    0    0    0    0    1    1    1
0    0    0    0    0    0    0    1    1    1
0    0    0    0    0    0    0    1    1    1
0    0    0    0    0    0    0    1    1    0
0    0    0    0    0    0    0    1    0    1
0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0
```

The graphical representation of this matrix is shown in Figure B-2. Values must be assigned to variables in the correct order to ensure each node's parents are instantiated before it is conditionally sampled. The ordering in the parameters file is already sorted because all directed edges originate from variables 1–7 and end at variables 8–10; therefore all parent variables 1–7 will be instantiated first, followed by child variables 8–10.

It is only necessary to assign new values to the dynamic variables, namely $\dot{v}(t+1)$, $\dot{h}(t+1)$ and $\dot{\psi}(t+1)$. The process is similar to that used to sample from the initial network. First, consult the table of sufficient statistics for $\dot{v}(t+1)$, which provides $N(\dot{v}(t+1) \mid L, v, \dot{v}(t), \dot{h}(t), \dot{\psi}(t))$. Sampling requires only the row representing the current variable assignment for the parents: $L = [500, 1200)$, $v = [120, 140)$, $\dot{v} = [-0.25, 0.25)$, $\dot{h} = [-2000, -1250)$, and $\dot{\psi} = [-1.5, 1.5)$. The conditional probability table is not reproduced here due to its size; a previous report includes a complete example of sampling a transition network [1].

Next, look at the row of interest and randomly assign a value to $\dot{v}(t+1)$ with probability proportional to the corresponding elements plus 1. Moving on to the other variables, assign $\dot{h}(t+1)$ to a random value conditional on A, L, v, $\dot{v}(t)$, and $\dot{h}(t)$. Finally, assign $\dot{\psi}(t+1)$ based on the assignments of A, L, v, $\dot{v}(t)$, and $\dot{\psi}(t)$. Repeat the process for each time step. The length of the
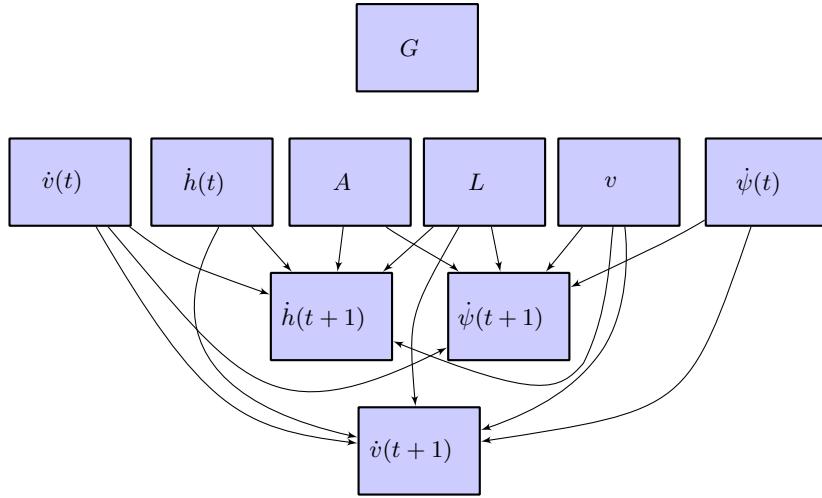
Figure B-2. A graphical representation of the transition network.

trajectory may in principle be as long as desired, though one must take care to verify that sampled trajectories longer than a few minutes have appropriate behavior; Appendix C discusses limitations due to the decrease in fidelity of feature distributions as trajectories are propagated.

Sampling from the Bayesian networks produces a sequence of assignments of variables to discrete bins such as $v = [120, 140)$. As Section 5 describes, the next step is to sample uniformly within the bins to produce continuous real values—with the exception of bins spanning 0, in which case 0 itself is selected. In cases where a bin does not change from one time step to the next, sampling within each bin at each time step would cause excessive variability in the vertical rates, turn rates, and acceleration. The process therefore only resamples within bins at mean rates specified in the parameters file. Typically, the variables continue in the same bin at the next time step.

The first four variables, G, A, L, and v, have zero as their resample rates because they are not resampled during the course of the trajectory. The last three elements specify the rates with which $\dot{v}$, $\dot{h}$, and $\dot{\psi}$ change within bins. As the trajectory evolves, it is necessary to sample within a bin whenever a variable switches to a new bin.

Once the initial conditions and a series of control variables ($\dot{v}$, $\dot{h}$, and $\dot{\psi}$) have been selected, the aircraft trajectory is constructed or simulated using an appropriate dynamic model. The control variables are assumed to be held constant during each 1-second time step.

54

APPENDIX C
MODEL VALIDITY LENGTH ASSESSMENT

It is helpful to investigate what types of SAA simulations the uncorrelated encounter model can support. One way to address this question is to assess the model validity length, or MVL, which is defined as the length of time that simulated tracks generated from the encounter model remain representative of aircraft behavior observed in the NAS. MVL is found to be finite because simulated tracks eventually become collectively unrepresentative of the airspace as they are propagated. More specifically, the distribution of track features eventually becomes very different from the initial, observed distribution.

Section C.1 discusses MVL in more detail and estimates a minimum MVL requirement for a self-separation encounter model. Section C.2 assesses the model against this requirement.

C.1   MODEL VALIDITY LENGTH DEFINITION AND REQUIREMENT

Figure C-1 illustrates the process used to determine the MVL, which is one measure of the stochastic stability or "validity" of the model. Validity is assessed by comparing the marginal feature distributions of sampled tracks—such as airspeed, acceleration and turn rate—to the observed distributions. This process starts by sampling tracks of length 600 seconds from the encounter model as described in Section 5. The tracks are propagated using a dynamic model and feature distributions at each time step are captured. Features are then discretized and counted into bins, with bin cutpoints defined as in the model. The result is a time history of the marginal, discrete feature distributions for the set of tracks, with a separate distribution for each variable in the model. These distributions are compared to those originally observed in the NAS. The first time step at which a variable's simulated distribution is no longer similar to the observed distribution is considered the MVL.

Airspeed in particular is important to analyze because it is not explicitly defined in the model's transition network, but rather is propagated in a dynamic simulation using the airspeed acceleration. Because the airspeed is the time integral of the acceleration, it is unbounded unless limits are enforced in the dynamic simulation environment.

Sections C.1.1, C.1.2, and C.1.3 provide more details of the process. Necessary background on the propagation of airspeed by the aircraft dynamic model is described first in Section C.1.1 followed by an explanation in Section C.1.2 of the analysis used to compare two distributions and decide when they are different. Section C.1.3 describes rejection sampling, a basic technique for extending MVL by rejecting unrealistic sampled tracks.

Section C.1.4 presents an estimate of the minimum time horizon or MVL needed to support simulation and test of SAA systems. This requirement is not well-defined but can be estimated as a function of the size of the encounter cylinder described in Section 6.
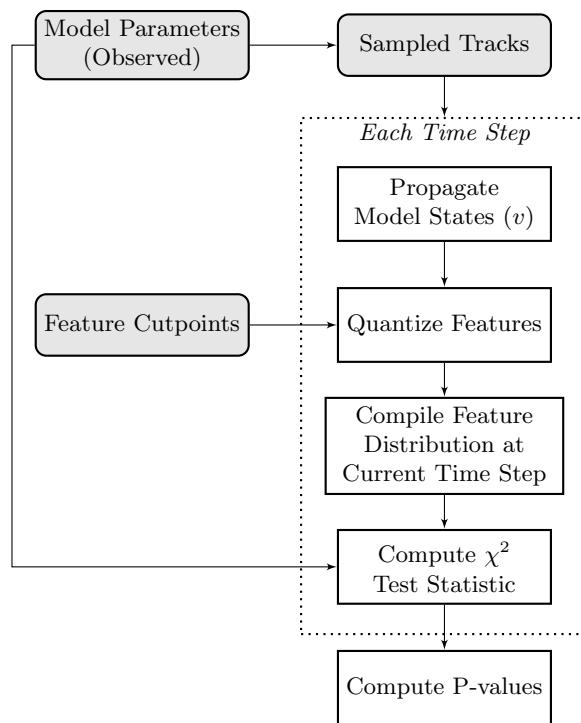
Figure C-1. Model validity length analysis approach. Both p-values and the actual distributions of airspeed and other model variables (acceleration, vertical rate, and turn rate) are compared.

C.1.1 Airspeed Propagation

Though airspeed is a parameter in the initial network, it is not explicitly updated in the transition network but rather propagated in the dynamic simulation by integrating the acceleration. Because airspeed is not explicitly bounded by the model, it must be bounded during the dynamic simulation because, for example, negative airspeed is not physically possible. Figure C-2 shows an example track where the airspeed is propagated to an unphysical state. In this case, the acceleration would cause a negative airspeed after about 400 s if left unbounded. The dynamic simulation bounds the airspeed at about 1 kt. The propagation of airspeed for this analysis is numerically approximated by the trapezoidal rule, or

$$v_i = v_{i-1} + (\dot{v}_{i-1} + \dot{v}_i)\Delta t/2 \,, \tag{C-1}$$

where the index i indicates the time step and Δt is the time step duration. The trapezoidal integral approximation with and without bound is shown in Figure C-2. The trapezoidal rule is modified for cases where the airspeed exceeds reasonable physical limits. If at a time step the airspeed is less than zero, it is bounded to zero during integration—for example, airspeeds propagated to negative values are replaced with zero. Likewise, if the airspeed is propagated to a value greater than the maximum sampled from the model, in this case 300 kt, then it is bounded to this maximum.
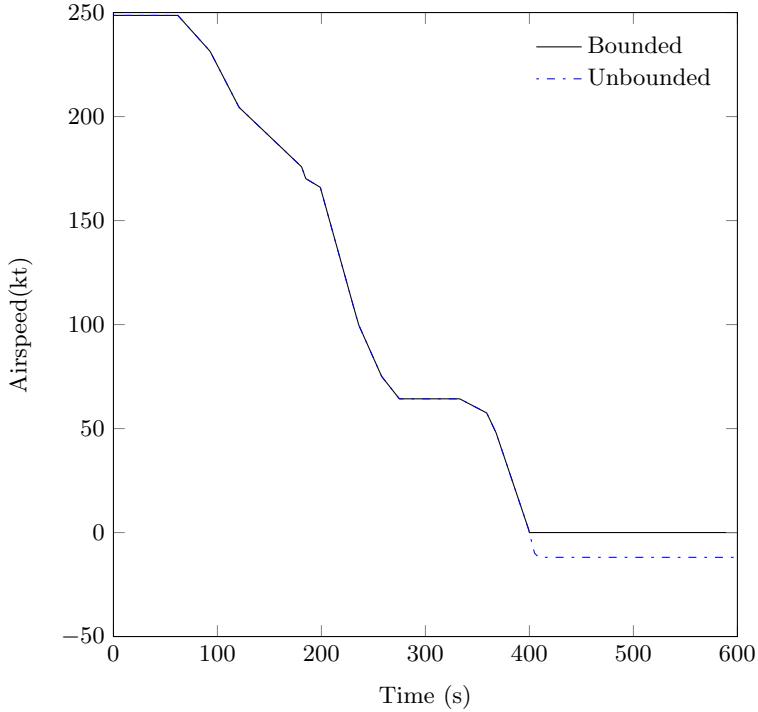

Figure C-2. Example trajectory propagated with bounded and unbounded trapezoidal integral approximations.

57

C.1.2 Distribution Comparison

As illustrated in Figure C-1, the simulated airspeed distribution for the set of sampled tracks is quantitatively compared at each time step to that observed in the airspace; the comparison uses a modified χ^2 goodness-of-fit test, which produces a p-value. Distributions for the other variables are compared in the same way.

The general goodness-of-fit test statistic can be described by

$$\chi^2 = \sum_{k=1}^{K} \frac{(n_{k,S} - n_{k,E})^2}{n_{k,E}} \,, \tag{C-2}$$

where K is the number of bins, n_k is the bin count, and the subscripts S and E denote simulated and expected, respectively; "expected" refers to the initial distribution observed in the NAS. The χ^2 distribution has $K - 1$ degrees of freedom. As the number of samples increases beyond about 10,000, χ^2 also increases so that minor deviations between two distributions may become statistically significant [27]. To overcome this, Hamada et al. suggest using a modified sample size N_m proportional to K, such as $K \approx N_m^{0.4}$ [28]. This method results in effective sample size $N_m = K^{2.5}$ and updates Equation C-2 to

$$\chi^2 = N_m \sum_{k=1}^{K} \frac{(f_{k,S} - f_{k,E})^2}{f_{k,E}} \,, \tag{C-3}$$

where $f = n/N$ and N is the original sample size.

The p-value resulting from the test is the probability of obtaining the sample's χ^2 test statistic or one more extreme by chance. The closer the p-value is to zero, the smaller the probability that the simulated distribution is similar to the expected distribution observed in the NAS.

Given the time history of p-values, the MVL is defined using a threshold p-value, or α value; the time at which the p-value decreases past this threshold is considered the MVL. Though the choice of α is somewhat arbitrary, p-values of 0.05 or 0.01 are typical choices; this work assumes $\alpha = 0.01$ or 1%. The p-value threshold corresponds to the maximum allowable Type-I error—the probability that the differences in the two distributions are due to chance alone. It is important to note that p-values resulting from statistical testing can be misleading and must be interpreted with care [27].

C.1.3 Rejection Sampling to Extend Model Validity Length

One simple approach to increasing MVL is to throw out samples with behavior like the track in Figure C-2, where the propagated airspeed decreases below zero or above the maximum modeled airspeed (300 kt) at any time during the track's duration. This is a rejection sampling method— a term which more generally describes any method of conditionally sampling a distribution or statistical model.

Though rejection sampling is a simple method, there are two major drawbacks. First, the distributions of other variables, such as acceleration, may change substantially due to rejection of extreme airspeeds. For example, the variance of the acceleration distribution may decrease because

removed samples have large accelerations that cause the airspeed to transition to an extreme state. Second, a large portion of the samples may need to be removed, which increases the computational cost of generating the samples.

C.1.4 Self-separation MVL Requirement

During initial encounter model development, an important requirement was that the models should provide an accurate representation of aircraft behavior for one minute near the closest point of approach (CPA). However, a self-separation encounter model will likely have a higher requirement due to the need for additional look-ahead time. This section quantitatively refines the requirement.

One approach to determining the MVL requirement is to assume a particular encounter cylinder size, simulate a large number of encounters, determine the time to closest point of approach (TCA) for each, and use the resulting distribution to define the requirement as sufficient to cover the majority of encounters. The requirement is defined here as the 95th percentile of the TCA distribution so that only 5% of encounters are expected to have TCA exceeding the required MVL; using this value instead of the maximum prevents outliers from distorting the result. Using TCA for this definition is preferable to using encounter length—the total time the intruder remains inside the encounter cylinder—because the time prior to CPA is more critical for avoidance maneuvering than the time after CPA.

However, a particular size assumption is not readily available because the encounter cylinder size for self-separation applications is not well-defined and may vary across applications. An alternate approach is to examine a variety of encounter cylinder sizes in order to estimate the relationship between cylinder size and the MVL requirement. This approach varies the cylinder radius from 2 nmi to 14 nmi in steps of 2 nmi while holding constant the assumptions for all other parameters. Intuitively, a linear relationship seems reasonable; it would imply that doubling the encounter cylinder size also doubles the MVL requirement. However, this assumption is not necessarily true, particularly if only the radius is varied and the height held constant.

Determining this relationship would allow the MVL requirement to be tailored to different self-separation applications which may assume different encounter cylinder sizes or shapes—for example, due to differences in sensor coverage area, field of regard or other factors. Smaller encounter cylinders should result in shorter encounters, hence relaxing demands on the encounter model; larger encounter cylinders would allow additional time to process avoidance logic and react to an encounter, hence relaxing demands on the SAA system. The choice of size for a particular application will involve a trade-off between these goals.

To determine the required MVL for each encounter cylinder, one million uncorrelated encounters are simulated using the encounter model and the TCA is captured for each encounter. There are several possible methods for defining the TCA; it is defined here as the time at which minimum horizontal separation occurs. Note that if a mean TCA across encounters is desired—not necessary in this case—the TCA for each encounter must be weighted appropriately in order to obtain a meaningful result.

To simplify the analysis, the trajectories of both aircraft are constrained to be non-maneuvering—that is, not turning or accelerating. Also, only encounters initialized on the side of the encounter cylinder are considered; top and bottom encounters are ignored. This simplifying assumption tends to increase the MVL requirement because TCA is typically larger for side encounters due to larger initial separation distance.

Figure C-3 plots the cumulative distribution function (CDF) of TCA for each cylinder radius; the horizontal line indicates the 95th percentile, which is chosen as the MVL requirement. Encounter cylinder height is fixed at 1500 ft in all cases. Table C-1 lists the 95% TCA for each encounter cylinder radius. This nearly linear relationship can be used to estimate the MVL requirement for a particular SAA application and, conversely, to estimate the maximum encounter cylinder size supported by an encounter model with a particular MVL.
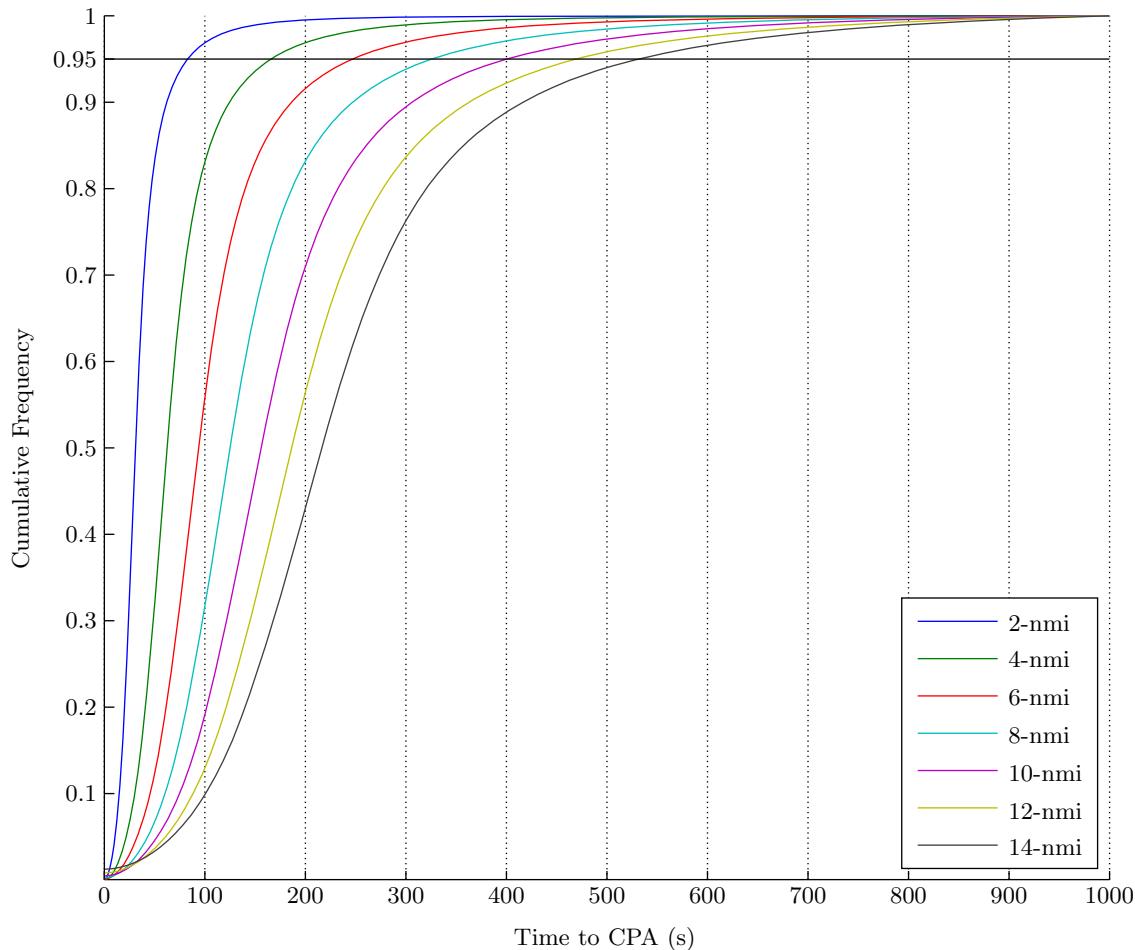

Figure C-3. Cumulative distribution of simulated encounter TCA (non-maneuvering trajectories and side encounters only); curves are labeled by encounter cylinder radius.

60

TABLE C-1

95% TCA (MVL Requirement) vs. encounter cylinder radius (nmi).

radius	95% TCA
2	82
4	165
6	246
8	325
10	400
12	468
14	531

The next section describes the results of applying the model validity length analysis to the uncorrelated encounter model.

C.2 MODEL VALIDITY LENGTH RESULTS

Section C.1 defined the the MVL and explored requirements for it. This section now evaluates the MVL for the model described in this report. It is important to understand the MVL because it is an indicator of what types of encounters and SAA applications the model can support.

MVL is evaluated for an encounter model using the method described in Section C.1, which applies a χ^2 test to a set of sampled tracks generated from the model; here the test uses a set of 100,000 sampled tracks, each 600 seconds long.

The MVL calculation yields the p-values in Figure C-4; the model has MVL of 166 seconds, which improves to 287 seconds with rejection sampling. Note that the maximum possible MVL in these results is 600 seconds because tracks sampled from the encounter model are 600 seconds long.

The p-value for the airspeed distribution is the limiting factor on the overall MVL since p-values for other features like acceleration and turn rate decay more slowly with simulation time. Figures C-5 and C-6 show that as sampled tracks are propagated, the airspeed distribution tends to spread out to extreme airspeeds near zero and near the maximum observed in the radar data. Note that rejection sampling yields a slower flattening of the airspeed distribution over time in Figure C-6, which is an improvement in the fidelity of the sampled tracks. Also note that the distribution at time 0 matches the observed distribution in Figure C-5, but the two distributions are slightly different in Figure C-6 due to the removal of rejection-sampled tracks.

Assuming MVL of 287 seconds, results from Section C.1.4 suggest the model could support encounter cylinders with radius between 6 and 8 nmi and possibly larger due to the conservative assumptions of that analysis.
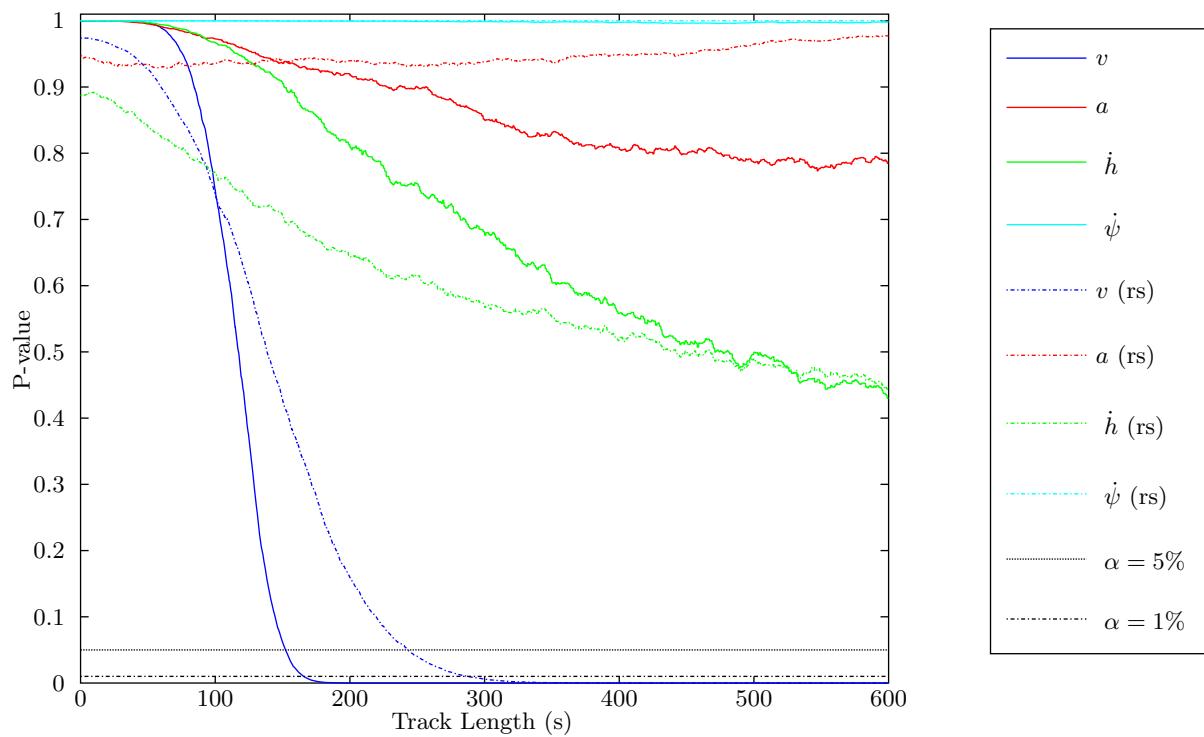
Figure C-4. P-value vs. time with rejection sampling for the uncorrelated encounter model; p-values for the rejection sampling method are indicated by "(rs)."
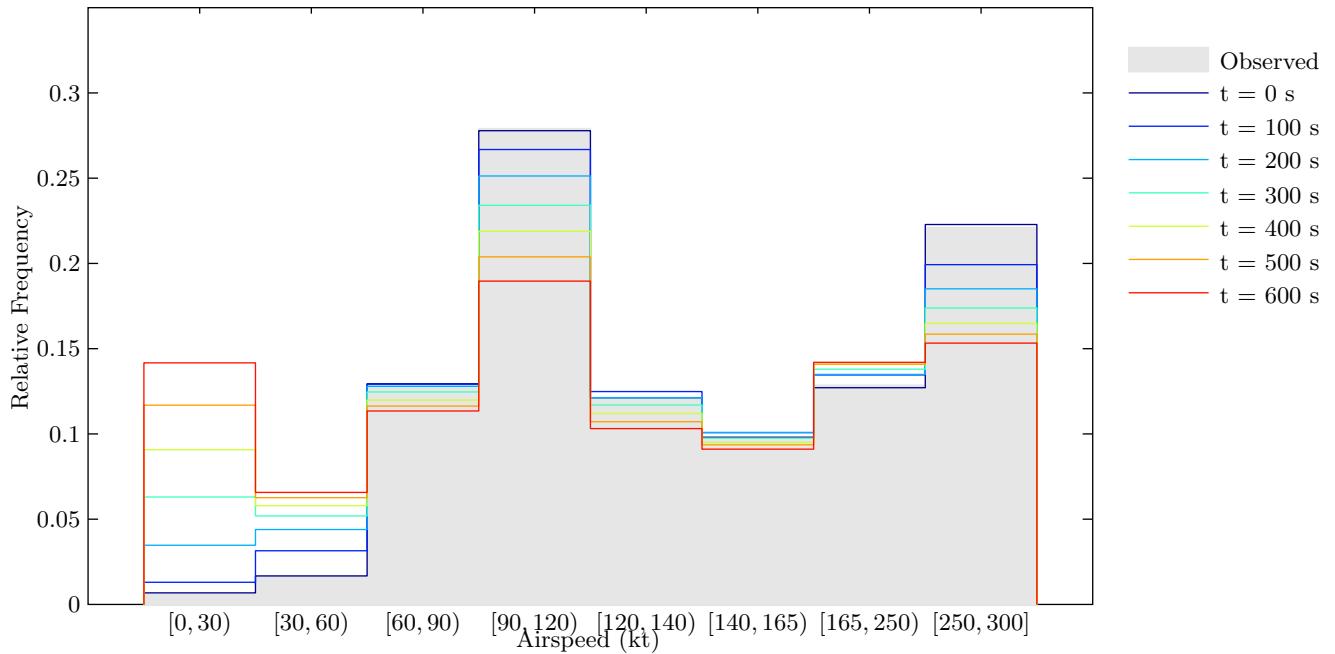
*Figure C-5. Simulated airspeed distribution at 100-second intervals.*
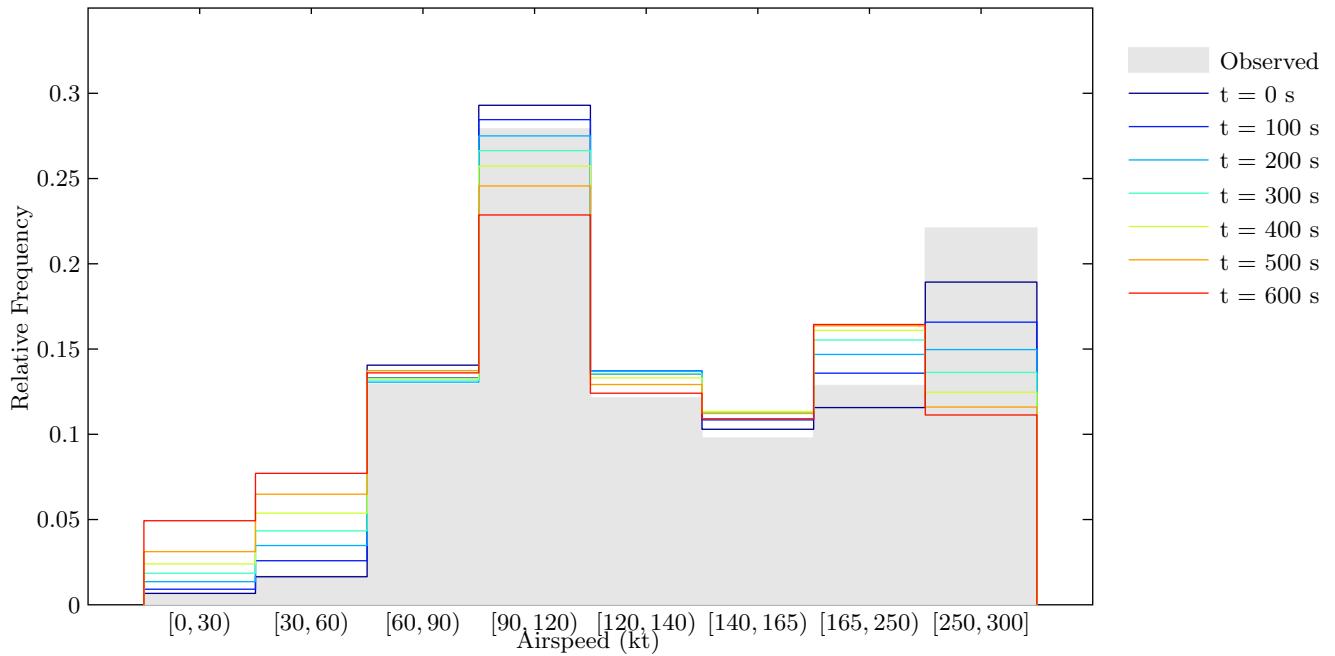
Figure C-6. Simulated airspeed distribution at 100-second intervals with rejection sampling.

63

This page intentionally left blank.

APPENDIX D
DYNAMIC SIMULATION VALIDATION

This appendix describes a set of sample trajectories that may be used as validation for those wishing to implement their own dynamic simulation models. A text file, named `uncor_tracks.txt` and available from Lincoln Laboratory, contains 50 synthetic tracks. Each track is 50 seconds in length and was generated by sampling model parameters from the dynamic Bayesian network and then simulating the aircraft trajectory in Lincoln Laboratory's dynamic simulation model.

The text file is a space-delimited list with the following ten columns:

- **ID:** Each row begins with an `id` number. All rows with the same `id` number correspond to a single track.

- **Time** t**:** Time values are reported once per second in the text file.

- **Vertical rate** $\dot{h}$**:** The vertical rate is reported in ft/min. This value is sampled from the dynamic Bayesian network.

- **Airspeed acceleration** $\dot{v}$**:** The airspeed acceleration is reported in kt/s and is also sampled from the dynamic Bayesian network.

- **Turn rate** $\dot{\psi}$**:** Turn rate is reported in deg/s and is also a variable from the dynamic Bayesian network.

- **Airspeed** v**:** The initial airspeed (kt) is defined by sampling from the initial Bayesian network. The remaining values are outputs from Lincoln Laboratory's dynamic simulation.

- **North position** x**:** Each aircraft is initialized at $x = 0$ ft.

- **East position** y**:** Each aircraft is initialized at $y = 0$ ft.

- **Altitude** h**:** All sample aircraft trajectories are initialized at an altitude of 10,000 ft.

- **Heading** ψ**:** All aircraft are initialized heading due north ($\psi = 0$ deg).

The dynamic variables ($\dot{h}$, $\dot{v}$, and $\dot{\psi}$) are inputs to an aircraft dynamic model that describe how the aircraft maneuvers throughout the simulation. The variables v, x, y, and h can be used to compare outputs of a simulation. If an aircraft is initialized at the origin with zero heading and an altitude of 10,000 ft, then the simulated track should approximately match the values provided in the text file.

Exactly matching simulation outputs with values in the text file is extremely unlikely due to the variety of acceptable aircraft models in simulation (e.g., degrees-of-freedom, transient dynamics, simulation step size, etc). The simulated and provided tracks only need to approximate each other in both the vertical and horizontal plane. Note that small differences in heading, resulting from a

turn early in the simulation, can result in large positional deviations further into the simulation—horizontal errors resulting from small differences in aircraft headings are acceptable for validating an aircraft dynamic simulation model and implementation of the uncorrelated encounter model.

Since there are numerous aircraft models that may be correctly implemented in simulation, this appendix does not define thresholds or minimum errors for validating a simulation. Instead, we only require that simulated aircraft must track steady-state inputs. For example, if sampling from the dynamic Bayesian network results in an aircraft flying straight (i.e., $\dot{\psi} = 0$ deg/s) that later transitions to $\dot{\psi} = 2$ deg/s, then the aircraft must eventually turn clockwise at 2 deg/s—the transient between $\dot{\psi} = 0$ and $\dot{\psi} = 2$ can differ from the sample trajectories.

APPENDIX E
TRACKING AND FUSION

Converting raw radar reports into tracks that are usable for our model development is a two-stage process. The first stage involves forming local tracks from the reports associated with each sensor. The second stage involves fusing the local tracks from multiple sensors to form global tracks. This appendix provides a brief overview of tracking and fusion.

We use the tracking algorithms from the Mode S and ASR-9 systems [19], the two most modern sensors in the Air Traffic Control System. The beacon correlation algorithms come from Mode S and the primary radar correlation algorithms come from ASR-9 Processor Augmentation Card (9-PAC). Both systems are integrated to provide a consistent track, although for the purpose of this report we ignore primary only reports.

After the reports of each sensor have been correlated into local tracks, we can fuse the local tracks to provide a global picture of the airspace. Fusion performs the following functions:

1. Merge tracks from multiple sensors that correspond to the same aircraft into a single global track.

2. Compute the speed and heading of each global track to permit trajectory predictions.

3. Correct sensor tracking errors that could lead to split global tracks and false encounters.

We use a *track-to-track* fusion method, meaning that we track each sensor's individual reports and then we merge all of the tracks [29]. The main advantages of *track-to-track* fusion over *report-to-track* fusion, in which all reports are correlated directly to global tracks, are:

1. Bias independence for velocity determination and maneuver detection.

2. Removal of short update interval velocity anomalies.

3. Reduced likelihood of forming clutter tracks.

4. Reduced likelihood of introducing incorrect data points into the track due to correlation errors.

Track merging employs position, velocity, Mode A code, and altitude as matching attributes over the entire track. Track merging for tracks with discrete codes, which are unique within an area, employs large correlation boxes for each of the matching attributes. Other tracks (1200 code or radar-only) must pass more stringent position tests and velocity tests in order to be merged together.

Our fusion method works forward in time. Thus, there is often doubt in whether or not tracks from multiple radars are indeed the same track with just a few data points. In cases of doubt, tentative matches are remembered that can be upgraded to a merge after more scans of

information are obtained. Merges are checked each scan and can be undone if later found to be unsatisfactory. Aircraft code changes are also accommodated, although they must be verified by other sensors in the merge set of the global track before being accepted. If only one sensor reports a code change, it is assumed that the sensor had a track swap, and the merge situation is altered accordingly.

The remainder of this section explains how we add local sensor tracks to global tracks, how to estimate track velocity as part of the fusion process, and how to filter for encounters.

E.1   ADDING LOCAL SENSOR TRACKS TO GLOBAL TRACKS

In order to facilitate fusing of tracks from multiple sensors into global tracks, we break the continental United States into 20 NM by 20 NM bins. Every track is associated to a geographic bin. Whenever the fusion process receives a new local sensor track, or a later report for an as yet unfused local track, an attempt is made to fuse this track to an existing global track. This process is performed by comparing the new track to all neighboring global tracks. The neighboring global tracks for a local track include all global tracks in the same geographical bin as the local track plus global tracks in the surrounding bins (totalling 9 bins). Several tests must be passed for the successful fusion of the single track to a global track:

1. The global track must not already be fused to another local track from the new track's sensor.

2. The tracks must agree on Mode A code (primary-only tracks automatically pass).

3. If the code agreement was on a discrete code, a very coarse horizontal positional test must be passed.

4. If the code agreement was on 1200 code, or no codes, a tighter positional test must be passed, as well as altitude and velocity tests. However, if only the velocity test fails, a potential fusion is declared; three successive potential fusions with the same global track results in a successful fusion.

If more than one possible global track satisfies the fusion tests, the one with the highest matching score is chosen. Existing fusion matches are checked each time a new report is received. If the tests fail for three scans in a row, the fusion is ended, and a new global track is sought for the local sensor track.

E.1.1   Code Matching Test

Normally, the code of the new track is the same as the code of the global track. However, code mismatching can complicate the fusion process. Code declaration errors due to data corruption, missing codes, and code changes due to controller action are all common. For this reason, associated with each global track is an established code and an alternate code, which is the code of the most recent report. Usually these two codes are the same. When an alternate code is different from the established code for three successive reports, however, the established code is updated to the

alternative code. Reports with no beacon code are ignored in this process. However, if a track has never been associated with a beacon code report, we consider this track to be a primary-only and give the track an established code of 0.

If both the local track the global track have a beacon code, then a successful code match is declared if any of the following statements are true:

1. The established codes match.

2. The alternate codes match.

3. One of the established codes and the other alternate code match.

However, failure is declared if the match is on code 0, and both tracks have a beacon code in the other code slot that do not match. Lastly, if one track is radar only, and the other track has a beacon code, failure is declared.

To handle local track code changes due to a track swap in the single sensor tracker, confirmation of the code change by the global track is required. If at the time of the local track code change, the global track has had an update by a different sensor's track with the old beacon code, a track swap is declared, and the local track is removed from fusion with the global track. The local track then undergoes a new fusion process.

E.1.2 Horizontal Position Matching Test

Horizontal positional matching requires agreement between the global track's most recent horizontal position $x_\mathrm{g}, y_\mathrm{g}$ and the new local track's horizontal position $x_\mathrm{l}, y_\mathrm{l}$ projected back to the time of the global track. This test is simple if both track's reports contained altitude. If a radar report contains altitude h, range ρ, azimuth θ, then the track's horizontal position x, y can be empirically determined.

If, however, the altitude of a track is unknown, then the tracks' altitude has to be assumed (or guessed) in order to derive the track's horizontal position. Simply guessing an altitude may produce a erroneous x, y position. In order to use a reasonable altitude value, we employ the following algorithm:

1. If only one track has known altitude, then we convert the other track's stored ρ, θ position to x, y using the first track's altitude.

2. If neither track has known altitude (which is always true for a primary-only match), we consider all altitudes from $0\,\mathrm{NM}$ to $7\,\mathrm{NM}$ at $1\,\mathrm{NM}$ steps. We then use the altitude that produces the closest positional match between the two tracks. While a smaller step size may produce more accurate estimates, we found that $1\,\mathrm{NM}$ is sufficient for fusing two tracks.

Horizontal positional agreement is declared if the horizontal distance between the two tracks is less than an acceptable value:

$$\sqrt{(x_g - x_l)^2 + (y_g - y_l)^2} \leq \Delta r_{max} + 3\sigma_{az}\rho_g + 3\sigma_{az}\rho_l, \tag{E-1}$$

where we use $\Delta r_{max} = 20\,\text{NM}$ for a discrete code match and $\Delta r_{max} = 1\,\text{NM}$ for a 1200 code or radar match. The standard deviation for horizontal position error σ_{az} accounts for positional errors due to azimuth noise in radar measurements, which is the dominant source of horizontal position error. We model the standard deviation of the azimuth noise as 3 milliradians for the data format of our radar feed.

E.1.3 Altitude Matching Test

Altitude matching requires agreement between the local and global track altitudes when both are known. Two comparisons are tested; the success of either test results in a match. The comparison test is:

$$
\begin{aligned}
\Delta t_l &= t_g - t_l \\
\Delta h_l &= |h_g - h_l| \\
\Delta h_l &\leq \Delta h_{max} \\
\frac{\Delta h_l}{\Delta t_l} &\leq \Delta \dot{h}_{max}
\end{aligned}
$$

where we use $\Delta h_{max} = 600\,\text{ft}$ and $\Delta \dot{h}_{max} = 100\,\text{ft/s}$. Since the altitude of a track can significantly change between sequential reports, we test both the most recent and previous local track altitudes with the most recent global track altitude update. Only one of the local track altitudes is required to pass the test.

E.1.4 Velocity Matching Test

Velocity matching requires agreement between the two track headings ψ and speeds s according to the following tests:

$$
\begin{aligned}
|\psi_g - \psi_l| &\leq \Delta \psi_{max} \\
|s_g - s_l| &\leq \Delta s_{max} \\
\frac{1}{2} \leq \frac{s_g}{s_l} &\leq 2
\end{aligned}
$$

where we use $\Delta \psi_{max} = 45°$ and $\Delta s_{max} = 100\,\text{kt}$. The last test is needed for slow aircraft and clutter tracks, to prevent, for example, speeds of 20 and 110 kt from agreeing.

E.2 DETERMINING TRACK AIRSPEED AND HEADING

Determining a global track's airspeed and heading is a two-step process. First, the individual sensor tracks are smoothed. Second, the individual tracks are averaged using relative weights that

account for sensor update times and the quality of each sensor's measurement. We apply both alpha smoothing and curve fitting to determine airspeed and heading, depending upon the track situation. We have implemented various maneuver detection algorithms, and tracking is dependent upon the current turn rate and acceleration states of the track.

E.2.1   Local Track Smoothing

First, we require that the track has moved a minimum distance for it to be considered. If the track never moves more than $1\,\mathrm{NM}$, then the track is thrown out. After the movement test is satisfied, the track's airspeed and heading are calculated from the new and previous positions. We then update the local track's airspeed and heading estimates using alpha smoothing.

First, the current heading estimate $\psi^{(j)}$, and its difference from the previous estimate $\psi^{(j-1)}$, are given by

$$
\begin{aligned}
\psi^{(j)} &= \mathrm{atan2}\left((x^{(j)} - x^{(j-1)}), (y^{(j)} - y^{(j-1)})\right) \\
\Delta\psi^{(j)} &= \psi^{(j)} - \psi^{(j-1)}
\end{aligned}
$$

Next, we determine the current turn rate state S_ψ of the track:

$$
S_\psi^{(n)} = \begin{cases}
2 & \text{if } \Delta\psi^{(j)} > \sigma_{\text{heading}} \\
1 & \text{if } \Delta\psi^{(j)} > \Delta\psi_{\min} \\
-2 & \text{if } \Delta\psi^{(j)} < -\sigma_{\text{heading}} \\
-1 & \text{if } \Delta\psi^{(j)} < -\Delta\psi_{\min} \\
0 & \text{otherwise}
\end{cases}
\tag{E-2}
$$

where we use $\Delta\psi_{\min} = 3°$ and σ_{heading} is the standard deviation of the heading noise, which is calculated from the standard deviations for range and azimuth noise of the sensors. Note that a positive $\Delta\psi$ value corresponds to a right turn, while a negative value corresponds to a left turn. We then use $S_\psi^{(j)}$ to determine the smoothing value alpha α in Table E-1 of the individual tracks that will be used to calculate the heading of the global track at the current time.

The new track heading is finally given by:

$$
\psi^{(j)} = \psi^{(j-1)} + \alpha \times \Delta\psi^{(j)}
$$

and we iterate through this process for the entire track.

The process to estimate airspeed s is similar, with one important difference. If successive positions are simply connected, then the airspeed estimates will always be too high, since the aircraft will appear to "zig-zag" along the track. Thus, only the projection of the velocity vector

Smoothing values depending on the current and previous turn states.

	Current Turn State				
Previous State	Large Left Turn (-2)	Small Left Turn (-1)	No Turn (0)	Small Right Turn (+1)	Large Right Turn (+2)
Large Left (-2)	0.7	0.7	0.4	0.5	0.5
Small Left (-1)	0.7	0.4	0.4	0.5	0.5
No Turn (0)	0.4	0.4	0.3	0.4	0.4
Small Right (+1)	0.5	0.5	0.4	0.4	0.7
Large Right (+2)	0.5	0.5	0.4	0.7	0.7

onto the track's heading vector is used to determine the track's airspeed:

$$s^{(j)} = \cos\left(\frac{\Delta\psi^{(j)}}{2}\right) \times \sqrt{\frac{(x^{(j)} - x^{(j-1)})^2 + (y^{(j)} - y^{(j-1)})^2}{t^{(j)} - t^{(j-1)}}}$$

$$\Delta s^{(j)} = s^{(j)} - s^{(j-1)}$$

We then determine the current airspeed acceleration state S_s of the track using a similar technique as we did for turn rate.

$$S_s^{(j)} = \begin{cases} 2 & \text{if } \Delta s^{(j)} > \sigma_{\text{speed}} \\ 1 & \text{if } \Delta s^{(j)} > \Delta s_{\text{min}} \\ -2 & \text{if } \Delta s^{(j)} < -\sigma_{\text{heading}} \\ -1 & \text{if } \Delta s^{(j)} < -\Delta s_{\text{min}} \\ 0 & \text{otherwise} \end{cases}$$

where we use $\Delta s_{\text{min}} = 18\,\text{kt}$ and σ_{speed} is the standard deviation of airspeed error due to noise in range and azimuth measurements from the radar sensors. The speed smoothing and the speed alpha table rules are the same as for the heading case.

E.2.2 Global Track Smoothing

In order to determine a global track's airspeed and heading at each measurement, we use a weighted least squares estimation approach. In this section we describe in detail the approach for determining the track's heading.

First, each sensor's heading estimate is assigned a weight w_i at the current time $t^{(c)}$ as follows:

$$w_i^{(c)} = \sigma_{\text{heading}}^{-1} \times \frac{t_{\text{max}} - \frac{t_i^{(j)} + t_i^{(j-1)}}{2} - t^{(c)}}{t_{\text{max}}}$$

where σ_{heading} is the standard deviation of the heading noise and $t_{\text{max}} = 18\,\text{s}$ is a discounting factor that takes into account the time difference between the measurement from the sensor being considered and the time for when we are determining heading. The time $t_i^{(j)}$ corresponds to the time of the next closest measurement for sensor i with respect to the current time that we are trying to determine the track's heading.

Next, we determine the total turn state score for the track

$$\left| \sum_{i=1}^{N} S_i \right| ,$$

where N is the number of sensors supporting the track and S_i is the current turn state value for the ith sensor defined in Equation E-2. If the turn rate score is less than N, then we consider the track to be non-turning and the current global heading is simply the weighted average of the N sensor heading estimates:

$$\psi_{\text{global}}^{(c)} = \frac{\sum_{i=1}^{N} \psi_i^{(j)} \times w_i^{(j)}}{\sum_{i=1}^{N} w_i^{(c)}}$$

Otherwise, if the turn rate score is greater than or equal to N, then we consider the track to be in a turn. In this case, we utilize weighted least squares estimated to determine a first-order relationship between time and heading.

The global track speed calculation is identical in form to the global track heading calculation.

This page intentionally left blank.

APPENDIX F
BAYESIAN NETWORKS

This appendix briefly reviews Bayesian networks. Further discussion of Bayesian networks may be found elsewhere [30–32].

F.1 DEFINITION

A Bayesian network is a graphical representation of a multivariate probability distribution over variables $\boldsymbol{X} = X_1, \ldots, X_n$. In particular, a Bayesian network is a directed acyclic graph G whose nodes correspond to variables and edges correspond to probabilistic dependencies between them. Associated with each variable X_i is a conditional probability distribution $P(x_i \mid \boldsymbol{\pi}_i)$, where $\boldsymbol{\pi}_i$ denotes an instantiation of the parents of X_i in the graph. The probability of an instantiation of the variables is specified directly by the conditional probability distributions in the Bayesian network:

$$P(\boldsymbol{x}) = P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid \boldsymbol{\pi}_i) \,. \tag{F-1}$$

F.2 SAMPLING

It is rather straightforward to sample from a multivariate distribution represented by a Bayesian network. The first step is to produce a topological sort of the nodes in the network. A topological sort orders the nodes in a Bayesian network such that if a node X_i comes before X_j there does not exist a directed path from X_j to X_i. Every Bayesian network has at least one topological sort, but there may be many. Efficient algorithms exist for finding a valid topological sort [33].

To produce a sample from the joint distribution represented by a Bayesian network, iterate through a topologically sorted sequence of the variables and sample from their conditional probability distributions. The topological sort ensures that when sampling from each conditional probability distribution the necessary parents have been instantiated.

F.3 PARAMETER LEARNING

The parameters $\boldsymbol{\theta}$ of a Bayesian network determine the associated conditional probability distributions. Given some fixed network structure G, it is possible to learn these parameters from data. This appendix assumes that the variables are discrete.

Before discussing how to learn the parameters of a Bayesian network, it is necessary to introduce some notation. Let r_i represent the number of instantiations of X_i and q_i represent the number of instantiations of the parents of X_i. If X_i has no parents, then $q_i = 1$. The jth instantiation of the parents of X_i is denoted $\boldsymbol{\pi}_{ij}$.

There are $\sum_{i=1}^{n} r_i q_i$ parameters in a Bayesian network. Each parameter is written θ_{ijk} and determines $P(X_i = k \mid \boldsymbol{\pi}_{ij})$, i.e.,

$$P(X_i = k \mid \boldsymbol{\pi}_{ij}) = \theta_{ijk}.$$

Although there are $\sum_{i=1}^{n} r_i q_i$ parameters, only $\sum_{i=1}^{n} (r_i - 1) q_i$ are independent.

Computing the posterior $p(\boldsymbol{\theta} \mid D, G)$ involves specifying a prior $p(\boldsymbol{\theta} \mid G)$ and applying Bayes' rule

$$p(\boldsymbol{\theta} \mid D, G) = \frac{P(D \mid \boldsymbol{\theta}, G) p(\boldsymbol{\theta} \mid G)}{P(D \mid G)} = \frac{P(D \mid \boldsymbol{\theta}, G) p(\boldsymbol{\theta} \mid G)}{\int P(D \mid \boldsymbol{\theta}, G) p(\boldsymbol{\theta} \mid G) \, d\boldsymbol{\theta}}. \tag{F-2}$$

If N_{ijk} is the count of $X_i = k$ given $\boldsymbol{\pi}_{ij}$ in the data D, then the probability of the data given the parameters $\boldsymbol{\theta}$ is

$$P(D \mid \boldsymbol{\theta}) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \theta_{ijk}^{N_{ijk}}. \tag{F-3}$$

Let $\boldsymbol{\theta}_{ij} = (\theta_{ij1}, \ldots, \theta_{ijr_i})$. Since $\boldsymbol{\theta}_{ij}$ is independent of $\boldsymbol{\theta}_{i'j'}$ when $ij \neq i'j'$, the prior probability of the parameters assuming a fixed structure G is

$$p(\boldsymbol{\theta} \mid G) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} p(\boldsymbol{\theta}_{ij} \mid G). \tag{F-4}$$

The density $p(\boldsymbol{\theta}_{ij} \mid G)$ is a distribution over relative frequencies. Under some very weak assumptions, it is possible to prove that $p(\boldsymbol{\theta}_{ij} \mid G)$ is Dirichlet (see [32], Section 6.2.3). Hence,

$$p(\boldsymbol{\theta}_{ij} \mid G) = \begin{cases} \frac{\Gamma(\alpha_{ij0})}{\prod_{k=1}^{r} \Gamma(\alpha_{ijk})} \prod_{k=1}^{r_i} \theta_{ijk}^{\alpha_{ijk}-1} & \text{if } 0 \leq \theta_{ijk} \leq 1 \text{ and } \sum_{k=1}^{r_i} \theta_{ijk} = 1 \\ 0 & \text{otherwise} \end{cases},$$

where $\alpha_{ij1}, \ldots, \alpha_{ijr_i}$ are the parameters of the Dirichlet distribution and $\alpha_{ij0} = \sum_{k=1}^{r_i} \alpha_{ijk}$. For the prior to be objective (or noninformative), the parameters α_{ijk} must be identical for all k. Different objective priors have been used in the literature. Cooper and Herskovits [26] use $\alpha_{ijk} = 1$. Heckerman, Geiger, and Chickering [34] use and justify $\alpha_{ijk} = 1/(r_i q_i)$.

It is possible to show that $p(\boldsymbol{\theta}_{ij} \mid D, G)$ is Dirichlet with parameters $\alpha_{ijk} + N_{ijk}, \ldots, \alpha_{ijk} + N_{ijk}$. Hence,

$$p(\boldsymbol{\theta}_{ij} \mid D, G) = \begin{cases} \frac{\Gamma(\alpha_{ij0}+N_{ij})}{\prod_{k=1}^{r} \Gamma(\alpha_{ijk}+N_{ijk})} \prod_{k=1}^{r_i} \theta_{ijk}^{\alpha_{ijk}+N_{ijk}-1} & \text{if } 0 \leq \theta_{ijk} \leq 1 \text{ and } \sum_{k=1}^{r_i} \theta_{ijk} = 1 \\ 0 & \text{otherwise} \end{cases},$$

where $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$.

Sampling from a Bayesian network with G known, $\boldsymbol{\theta}$ unknown, and D observed involves assigning k to X_i with probability

$$P(X_i = k \mid \pi_{ij}, D, G) = \int \theta_{ijk} p(\theta_{ijk} \mid D, G) \, d\theta_{ijk} = \frac{\alpha_{ijk} + N_{ijk}}{\sum_{k'=1}^{r_i} (\alpha_{ijk'} + N_{ijk'})}. \tag{F-5}$$

F.4 STRUCTURE LEARNING

This section discusses finding the most likely structure G that generated the observed set of data D. By Bayes' rule,

$$P(G \mid D) \propto P(G)P(D \mid G) = P(G) \int P(D \mid \boldsymbol{\theta}, G)p(\boldsymbol{\theta} \mid G)\, \mathrm{d}\boldsymbol{\theta}. \tag{F-6}$$

The previous section explains how to compute the likelihood $P(D \mid \boldsymbol{\theta}, G)$ and the prior $p(\boldsymbol{\theta} \mid G)$. Cooper and Herskovits [26] show how to evaluate the integral above, resulting in

$$P(G \mid D) = P(G) \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij0})}{\Gamma(\alpha_{ij0} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}, \tag{F-7}$$

where $N_{ij} = \sum_{k=1}^{r} N_{ijk}$. Heckerman, Geiger, and Chickering [34] suggest priors over graphs, but it is not uncommon in the literature to assume a uniform prior. For numerical convenience, most Bayesian network learning packages calculate and report $\log P(G \mid D) + K$, where K is a constant independent of G. This quantity is often called the *Bayesian score* and may be used for structure comparison and search.

This page intentionally left blank.

REFERENCES

[1] M. Kochenderfer, J. Kuchar, L. Espindle, and J. Griffith, "Uncorrelated encounter model of the National Airspace System version 1.0," MIT Lincoln Laboratory, Lexington, Massachusetts, Project Report ATC-345 (2008).

[2] MITRE, "System safety study of minimum TCAS II," MITRE, Technical Rep. MTR-83W241 (1983).

[3] A. Drumm, "Lincoln Laboratory evaluation of TCAS II Logic Version 6.04a," MIT Lincoln Laboratory, Project Report ATC-240 (1996).

[4] M.P. McLaughlin, "Safety study of the Traffic Alert and Collision Avoidance System (TCAS II)," MITRE Corporation, Technical Rep. MTR 97W32 (1997).

[5] B. Chludzinski, "Lincoln Laboratory evaluation of TCAS II logic version 7," MIT Lincoln Laboratory, Project Report ATC-268 (1999).

[6] T. Arino, K. Carpenter, S. Chabert, H. Hutchinson, T. Miquel, B. Raynaud, K. Rigotti, and E. Vallauri, "Studies on the safety of ACAS II in Europe," Eurocontrol, Technical Rep. ACASA/WP-1.8/210D (2002).

[7] ICAO, "ACAS manual," Eurocontrol, Technical Rep. SCRSP/1-WP/53 (2004).

[8] T.A. Choyce and K.M. Ciaramella, "Test and evaluation of TCAS II logic version 7," Federal Aviation Administration, Technical rep. (2000).

[9] ICAO, "Surveillance, radar and collision avoidance," in *ICAO Standards and Recommended Practices*, vol. IV, annex 10 (1998).

[10] T. Miquel and K. Rigotti, "European encounter model," CENA/Sofréavia and QinetiQ, Technical Rep. ACASA/WP1/186/D (2002).

[11] S. Chabert, "Safety encounter model focused on issue SA01a," CENA/Sofréavia and QinetiQ, Technical Rep. SIRE/WP2/21/D (2005).

[12] M. Edwards, M. Kochenderfer, J. Kuchar, and L. Espindle, "Encounter models for unconventional aircraft," MIT Lincoln Laboratory, Project Report ATC-348 (2009).

[13] "Unmanned systems roadmap: 2007 - 2032," Department of Defense, Technical rep. (2007).

[14] M.W.M. Edwards, "Encounter models for the littoral regions of the national airspace system," MIT Lincoln Laboratory, Project Report CASSATT-2 (2010).

[15] M. Kochenderfer, L. Espindle, J. Kuchar, and J. Griffith, "Correlated encounter model for cooperative aircraft in the National Airspace System version 1.0," MIT Lincoln Laboratory, Lexington, Massachusetts, Project Report ATC-344 (2008).

[16] W.M. Bolstad, *Introduction to Bayesian Statistics*, Wiley, 2nd ed. (2007).

[17] T. Dean and K. Kanazawa, "A model for reasoning about persistence and causation," *Computational Intelligence* 5(3), 142–150 (1989).

[18] K. Murphy, *Dynamic Bayesian Networks: Representation, Inference and Learning*, Ph.D. thesis, University of California, Berkeley (2002).

[19] R.D. Grappel, "ASR-9 Processor Augmentation Card (9-PAC) phase II scan-scan correlator algorithms," MIT Lincoln Laboratory, Project Report ATC-298 (2001).

[20] J.L. Gertz, "Mode S surveillance netting," MIT Lincoln Laboratory, Project Report ATC-120 (1983).

[21] F.N. Fritsch and R.E. Carlson, "Monotone piecewise cubic interpolation," *SIAM Journal of Numerical Analysis* 17(2), 238–246 (1980).

[22] E.M. Shank, "A coordinate conversion algorithm for multisensor data processing," MIT Lincoln Laboratory, Project Report ATC-139 (1986).

[23] L.V. Schmidt, *Introduction to aircraft flight dynamics*, Reston, VA: American Institute of Aeronautics and Astronautics (1998).

[24] E. Maki, A. Weinert, and M. Kochenderfer, "Efficiently estimating ambient near mid-air collision risk for unmanned aircraft," in *AIAA Aviation Technology, Integration, and Operations Conference*, Fort Worth, TX (2010).

[25] R. Srinivasan, *Importance Sampling: Applications in Communications and Detection*, Springer (2002).

[26] G.F. Cooper and E. Herskovits, "A Bayesian method for the induction of probabilistic networks from data," *Machine Learning* 9(4), 309–347 (1992).

[27] D.H. Johnson, "The insignificance of statistical significance testing," *The Journal of Wildlife Management* 63(3), 763–772 (1999), URL http://www.jstor.org/stable/3802789.

[28] M. Hamada, A. Wilson, C. Reese, and H. Martz, *Bayesian Reliability*, Springer (2008).

[29] J.L. Gertz and R.D. Grappel, "Surveillance improvement algorithms for Airport Surface Detection Equipment Model X (ASDE-X) at Dallas-Fort Worth Airport," MIT Lincoln Laboratory, Project Report ATC-333 (2007).

[30] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, San Francisco, CA: Morgan Kaufmann (1988).

[31] F.V. Jensen, *Bayesian Networks and Decision Graphs*, Springer Verlag (2001).

[32] R.E. Neapolitan, *Learning Bayesian Networks*, Upper Saddle River, NJ: Prentice Hall (2004).

[33] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, 2nd ed. (2001).

[34] D. Heckerman, D. Geiger, and D.M. Chickering, "Learning Bayesian networks: The combination of knowledge and statistical data," *Machine Learning* 20(3), 197–243 (1995).

This page intentionally left blank.

REPORT DOCUMENTATION PAGE

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
19 August 2013	Project Report	

4. TITLE AND SUBTITLE

Uncorrelated Encounter Model of the National Airspace System, Version 2.0

5a. CONTRACT NUMBER
FA8721-05-C-0002

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Andrew J. Weinert, Eric P. Harkleroad, J. Daniel Griffith, Matthew W. Edwards, and Mykel J. Kochenderfer

5d. PROJECT NUMBER
2136

5e. TASK NUMBER
1

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

MIT Lincoln Laboratory
244 Wood Street
Lexington, MA 02420-9108

8. PERFORMING ORGANIZATION REPORT NUMBER

ATC-404

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

M. Viva Austin
UAS Project Office, SFAE-AV-UAS-TIM
Building 5300
Redstone Arsenal, AL 35898

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

ESC-EN-HA-TR-2012-124

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Airspace encounter models, which provide realistic close encounter situations representative of the airspace, are a critical component in the safety assessment of sense and avoid systems. Of particular relevance to Unmanned Aircraft Systems is the potential for encountering general aviation aircraft that are flying under Visual Flight Rules and which may not be in contact with air traffic control. In response to the need to develop a model of these types of encounters, Lincoln Laboratory undertook an extensive radar data collection and modeling effort involving over 200 radars across the United States. This report describes the structure, content, and usage of that encounter model. The model is based on the use of Bayesian networks to represent relationships between dynamic variables and to construct random aircraft trajectories that are statistically similar to those observed in the radar data. The result is a framework from which representative intruder trajectories can be generated and used in fast-time Monte Carlo simulations to provide accurate estimates of collision risk.

This report includes updates to the scope and data processing methods used for the initial version of the uncorrelated encounter model. The updates are motivated by increasing interest in sense and avoid applications related to self-separation in addition to collision avoidance. Self-separation encounter models are likely to have stricter requirements, including the need to simulate longer encounters and greater look-ahead time. Another enhancement to the model is an additional discrete variable specifying the geographic location, which allows a single model to provide specialized coverage of different geographic regions. These regions include the contiguous United States, Alaska, islands, and offshore areas.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT Unclassified	**b. ABSTRACT** Unclassified	**c. THIS PAGE** Unclassified	Same as report	93	**19b. TELEPHONE NUMBER** *(include area code)*